

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>7</sup> : G06F 17/14		A2	(11) International Publication Number: WO 00/51014
			(43) International Publication Date: 31 August 2000 (31.08.00)
(21) International Application Number: PCT/US00/04868		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 25 February 2000 (25.02.00)			
(30) Priority Data: 09/259,669 26 February 1999 (26.02.99) US 09/305,690 5 May 1999 (05.05.99) US			
(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052 (US).			
(72) Inventor: MALVAR, Henrique, S.; 2302 233rd Ave. NE, Redmond, WA 98053 (US).			
(74) Agent: DEFRANK, Edmond, A.; Lyon, Harr & Defrank, Suite 800, 300 East Esplanade Drive, Oxnard, CA 93030 (US).			

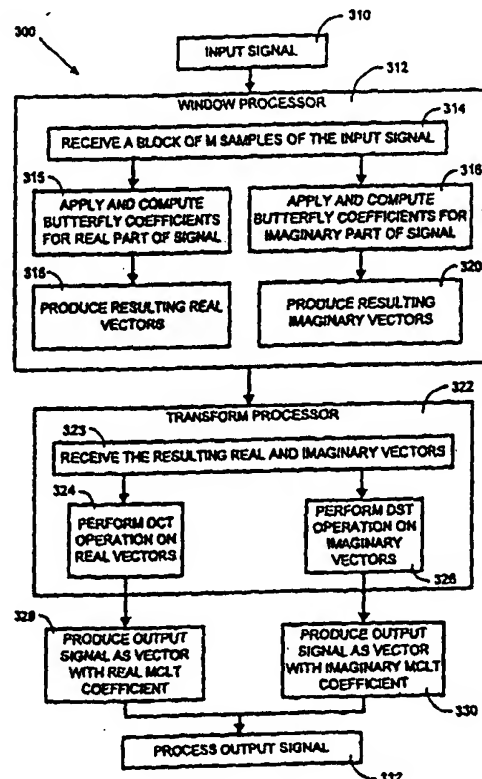
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: MODULATED COMPLEX LAPPED TRANSFORM FOR INTEGRATED SIGNAL ENHANCEMENT AND CODING

(57) Abstract

The present invention is embodied in a system and method for performing spectral analysis (1500) of a digital signal having a discrete duration by spectrally decomposing the digital signal at predefined frequencies uniformly distributed over a sampling frequency interval into complex frequency coefficients (220, 222) so that magnitude and phase information at each frequency is immediately available to produce a modulated complex lapped transform (MCLT). The present invention includes a MCLT processor (1510), an acoustic echo cancellation device (1512) and a noise reducer (1514) integrated with an encoder/decoder device (1500).



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

# MODULATED COMPLEX LAPPED TRANSFORM FOR INTEGRATED SIGNAL ENHANCEMENT AND CODING

## TECHNICAL FIELD

5           The present invention relates to a system and method for producing modulated complex lapped transforms (MCLTs), and in particular, a system and method for incorporating complex coefficients to modulated lapped transforms (MLTs) to derive MCLTs.

## 10       BACKGROUND ART

          In many engineering and scientific applications, it is desirable to analyze a signal in the frequency domain or represent the signal as a linear superposition of various sinusoids. The analysis of the amplitudes and phases of such sinusoids (the signal spectrum) can be useful in multimedia applications  
15       for operations such as noise reduction, compression, and pattern recognition, among other things. The Fourier transform is a classical tool used for frequency decomposition of a signal. The Fourier transform breaks a signal down to component frequencies. However, its usefulness is limited to signals that are stationary, i.e., spectral patterns of signals that do not change appreciably with  
20       time. Since most real-world signals, such as audio and video signals, are not stationary signals, localized frequency decompositions are used, such as time-frequency transforms. These transforms provide spectral information that is localized in time.

          One such transform is the discrete cosine transform (DCT). The DCT  
25       breaks a signal down to component frequencies. For instance, a block of  $M$  samples of the signal can be mapped to a block of  $M$  frequency components via a matrix of  $M \times M$  coefficients. To ensure a good energy compaction performance, the DCT approximates the eigenvectors of the autocorrelation matrix of typical signal blocks. Basis functions for the DCT (for type II) can be  
30       defined as:

$$a_{nk} = c(k) \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{1}{2} \right) \frac{k\pi}{M} \right]$$

where,  $a_{nk}$  is the element of an  $A$  transformation matrix in the  $n$ th row and  $k$ th column, or equivalently, the  $n$ th sample of the  $k$ th basis function. For

orthonormality, the scaling factors are chosen as:

$$c(k) = \begin{cases} 1/\sqrt{2} & \text{if } k = 0 \\ 1 & \text{otherwise} \end{cases}$$

- 5 The transform coefficients  $X(k)$  are computed from the signal block samples  $x(n)$  by:

$$X(k) = \sum_{n=0}^{M-1} a_{nk} x(n)$$

- The DCT can be used for convolution and correlation, because it satisfies a modified shift property. Typical uses of the DCT are in transform coding,  
10 spectral analysis, and frequency-domain adaptive filtering.

An alternative transform for spectral analysis is the discrete cosine transform, type IV (DCT-IV). The DCT-IV is obtained by shifting the frequencies of the DCT basis functions in eqn. (A) by  $\pi/2M$ , in the form:

$$a_{nk} = \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right]$$

- 15 Unlike the DCT, the scaling factor is identical for all basis functions. It should be noted that the DCT-IV basis functions have a frequency shift, when compared to the DCT basis. Nevertheless, these transforms still lead to orthogonal basis.

- The DCT and DCT-IV are useful tools for frequency-domain signal  
20 decomposition. However, they suffer from blocking artifacts. In typical applications, the transform coefficients  $X(k)$  are processed in some desired way: quantization, filtering, noise reduction, etc. Reconstructed signal blocks are obtained by applying the inverse transform to such modified coefficients. When such reconstructed signal blocks are pasted together to form the  
25 reconstructed signal (e.g. a decoded audio or video signal), there will be discontinuities at the block boundaries.

The modulated lapped transform (MLT) eliminates such discontinuities. The MLT is a particular form of a cosine-modulated filter bank that allows for perfect reconstruction. For example, a signal can be recovered exactly from its MLT coefficients. Also, the MLT does not have blocking artifacts, namely, the MLT provides a reconstructed signal that decays smoothly to zero at its boundaries, avoiding discontinuities along block boundaries. In addition, the MLT has almost optimal performance for transform coding of a wide variety of signals. Because of these properties, the MLT is being used in many applications, such as many modern audio and video coding systems, including Dolby AC-3, MPEG-2 Layer III, and others.

However, one disadvantage of the MLT for some applications is that its transform coefficients are real, and so they do not explicitly carry phase information. In some multimedia applications, such as audio processing, complex subbands are typically needed by noise reduction devices, via spectral subtraction, and acoustic echo cancellation devices. Namely, in many audio processing applications digital audio representations are commonplace. For example, music compact discs (CDs), Internet audio clips, satellite television, digital video discs (DVDs), and telephony (wired or cellular) rely on digital audio techniques.

Digital representation of an audio signal is achieved by converting the analog audio signal into a digital signal with an analog-to-digital (A/D) converter. The digital representation can then be encoded, compressed, stored, transferred, utilized, etc. The digital signal can then be converted back to an analog signal with a digital-to-analog (D/A) converter, if desired. The A/D and D/A converters sample the analog signal periodically, usually at one of the following standard frequencies: 8 kHz for telephony, Internet, videoconferencing; 11.025 kHz for Internet, CD-ROMs, 16 kHz for videoconferencing, long-distance audio broadcasting, Internet, future telephony; 22.05 kHz for CD-ROMs, Internet; 32 kHz for CD-ROMs, videoconferencing, ISDN audio; 44.1 kHz for Audio CDs; and 48 kHz for Studio audio production.

Typically, if the audio signal is to be encoded or compressed after conversion, raw bits produced by the A/D are usually formatted at 16 bits per

audio sample. For audio CDs, for example, the raw bit rate is  $44.1 \text{ kHz} \times 16 \text{ bits/sample} = 705.6 \text{ kbps}$  (kilobits per second). For telephony, the raw rate is  $8 \text{ kHz} \times 8 \text{ bits/sample} = 64 \text{ kbps}$ . For audio CDs, where the storage capacity is about 700 megabytes (5,600 megabits), the raw bits can be stored, and there is  
5 no need for compression. MiniDiscs, however, can only store about 140 megabytes, and so a compression of about 4:1 is necessary to fit 30 min to 1 hour of audio in a 2.5" MiniDisc.

For Internet telephony and most other applications, the raw bit rate is too high for most current channel capacities. As such, an efficient encoder/decoder  
10 (commonly referred to as coder/decoder, or codec) with good compressions is used. For example, for Internet telephony, the raw bit rate is 64 kbps, but the desired channel rate varies between 5 and 10 kbps. Therefore, a codec needs to compress the bit rate by a factor between 5 and 15, with minimum loss of perceived audio signal quality.

15 With the recent advances in processing chips, codecs can be implemented either in dedicated hardware, typically with programmable digital signal processor (DSP) chips, or in software in a general-purpose computer. Currently, commercial systems use many different digital audio technologies. Some examples include: ITU-T standards: G.711, G.726, G.722, G.728,  
20 G.723.1, and G.729; other telephony standards: GSM, half-rate GSM, cellular CDMA (IS-733); high-fidelity audio: Dolby AC-2 and AC-3, MPEG LII and LIII, Sony MiniDisc; Internet audio: ACELP-Net, DolbyNet, PictureTel Siren, RealAudio; and military applications: LPC-10 and USFS-1016 vocoders.

It is desirable to have codecs that can achieve low computational  
25 complexity and exhibit robustness to signal variations for allowing the codec to handle wider range of signals, i.e., the audio signals can be clean speech, noisy speech, multiple talkers, music, etc. without unduly compromising performance. Therefore what is needed is a new audio processing system that integrates an acoustic echo cancellation device and noise reducer with a codec for improving  
30 performance, reducing computational complexity, and reducing memory usage and processing delay. Whatever the merits of the above mentioned systems and methods, they do not achieve the benefits of the present invention.

DISCLOSURE OF THE INVENTION

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, the present invention is embodied in a system and method for performing spectral analysis of a digital signal having a discrete duration. The present invention performs spectral analysis by spectrally decomposing the digital signal at predefined frequencies uniformly distributed over a sampling frequency interval into complex frequency coefficients so that magnitude and phase information at each frequency is immediately available.

Namely, the system of the present invention produces a modulated complex lapped transform (MCLT) and includes real and imaginary window processors and real and imaginary transform processors. Each window processor has window functions and operators. The real window processor receives the input signal as sample blocks and applies and computes butterfly coefficients for the real part of the signal to produce resulting real vectors. The imaginary window processor receives the input signal as sample blocks and applies and computes butterfly coefficients for the imaginary part of the signal to produce resulting imaginary vectors. The real transform processor computes a spatial transform on the real vectors to produce a real transform coefficient for the MCLT. The imaginary transform processor computes a spatial transform on the imaginary vectors to produce an imaginary transform coefficient for the MCLT.

In addition, the system can include inverse transform module for inverse transformation of the encoded output. The inverse transform module can include components that are the exact inverse of the inverse real and imaginary transform processors and the real and imaginary inverse window processors. The encoded output is received and processed by inverse real and imaginary transform processors, and then received and processed by real and imaginary inverse window processors to produce an output signal that substantially matches the input signal.

The foregoing and still further features and advantages of the present invention as well as a more complete understanding thereof will be made apparent from a study of the following detailed description of the invention in connection with the accompanying drawings and appended claims.

5

#### BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a block diagram illustrating an apparatus for carrying out the invention;

10

FIG. 2 is a general block diagram illustrating a system for computing and encoding modulated complex lapped transforms in accordance with the present invention;

FIG. 3 is a general block/flow diagram illustrating a system and method for computing modulated lapped transforms in accordance with the present invention;

15

FIG. 4 is a detailed block/flow diagram illustrating computation of a modulated complex lapped transform in accordance with the present invention;

FIGS. 5A and 5B are detailed diagrams illustrating the window operation of the modulated complex lapped transform of FIG. 4;

20

FIG. 6 is a flow diagram illustrating operational computation of a modulated complex lapped transform in accordance with the present invention;

FIG. 7 is a general block diagram of a full-band adaptive filter;

FIG. 8 is a general block diagram of a frequency-domain MCLT-based adaptive filter in accordance with the present invention;

25

FIG. 9 is a block diagram of a working example of the adaptive filter of FIG. 8 of the present invention in the form of an acoustic echo cancellation device;

FIG. 10 is a general block diagram of an acoustic echo cancellation device with MCLT-based adaptive filters in accordance with the present invention;

FIG. 11 is a wave signal illustrating sample results of the working example of FIG. 9;

30

FIG. 12 is a general block diagram of a noise reduction device with MCLT-



based adaptive filters in accordance with the present invention.

FIG. 13 is a flow diagram illustrating operational computation of a noise reduction device incorporating the modulated complex lapped transform of the present invention; and

5        FIG. 14 is a wave signal illustrating sample results of the working example of FIGS. 12-13.

FIG. 15 is a block diagram of a working system of the present invention illustrated as an integrated signal enhancer and noise reducer with a codec.

## 10        BEST MODE FOR CARRYING OUT THE INVENTION

In the following description of the invention, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration a specific example in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

### Introduction:

The MCLT of the present invention can achieve short-time spectral decomposition of signals with explicit magnitude and phase information and perfect signal reconstruction. For instance, the MCLT of the present invention can use sine functions at defined frequencies and phases to generate an additional orthogonal decomposition. The defined frequencies and phases are preferably the same that the MLT basis functions use for cosine modulation of a particular window function with certain properties.

25        In addition, the MCLT of the present invention is easily integrated with MLT-based systems. Once the MCLT of a signal has been computed, its MLT can be trivially obtained simply by discarding the imaginary parts. The present invention can use both the cosine and sine modulating functions for producing a frame decomposition with desirable properties. Further, the cosine and sine  
30        modulations can be used to compute the real and imaginary parts of a transform that has all the magnitude/phase properties of the short-time Fourier transform, while allowing for perfect signal reconstruction. Consequently, the

novel MCLT of the present invention can be used in applications such as high-fidelity audio coding, adaptive filtering, acoustic echo cancellation, noise reduction, or any other application where high-fidelity signal reconstruction is required.

5

#### Exemplary Operating Environment

FIG. 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with a variety of computer system configurations, including personal computers, server computers, hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located on both local and remote computer storage media including memory storage devices.

With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional computer 100, including a processing unit 102, a system memory 104, and a system bus 106 that couples various system components including the system memory 104 to the processing unit 102. The system bus 106 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes computer storage media in the form of read only memory (ROM) 110 and random access memory (RAM) 112. A basic input/output system 114 (BIOS), containing the basic routines that helps to

transfer information between elements within computer 100, such as during start-up, is stored in ROM 110. The computer 100 may include a hard disk drive 116 for reading from and writing to a hard disk, not shown, a magnetic disk drive 118 for reading from or writing to a removable magnetic disk 120, and an optical disk drive 122 for reading from or writing to a removable optical disk 124 such as a CD ROM or other optical media. The hard disk drive 116, magnetic disk drive 128, and optical disk drive 122 are connected to the system bus 106 by a hard disk drive interface 126, a magnetic disk drive interface 128, and an optical drive interface 130, respectively. The drives and their associated computer-readable media provide storage of computer readable instructions, data structures, program modules and other data for the computer 100. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 120 and a removable optical disk 130, it should be appreciated by those skilled in the art that other types of computer readable media can store data that is accessible by a computer. Such computer readable media can be any available media that can be accessed by computer 100. By way of example, and not limitation, such computer readable media may comprise communication media and computer storage media. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set of changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as wired network or direct wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. By way of example, and not limitation, communication media includes wired media such as a wired network or direct wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Computer storage media includes any method or technology for the storage of information such as computer readable instructions, data structures, program modules or other data. By way of example, such storage media includes RAM, ROM, EPROM, flash memory or other memory technology, CD-ROM, digital

video disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 100. Combinations of any of the above should also be included within the scope of computer readable media.

A number of program modules may be stored on the hard disk, magnetic disk 120, optical disk 124, ROM 110 or RAM 112, including an operating system 132, one or more application programs 134, other program modules 136, and program data 138. A user may enter commands and information into the computer 100 through input devices such as a keyboard 140 and pointing device 142. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 102 through a serial port interface 144 that is coupled to the system bus 106, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 146 or other type of display device is also connected to the system bus 106 via an interface, such as a video adapter 148. In addition to the monitor 146, computers may also include other peripheral output devices (not shown), such as speakers and printers.

The computer 100 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 150. The remote computer 150 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 100, although only a memory storage device 152 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 154 and a wide area network (WAN) 156. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and Internet.

When used in a LAN networking environment, the computer 100 is connected to the local network 154 through a network interface or adapter 158. When used in a WAN networking environment, the computer 100 typically

includes a modem 160 or other means for establishing communications over the wide area network 156, such as the Internet. The modem 160, which may be internal or external, is connected to the system bus 106 via the serial port interface 144. In a networked environment, program modules depicted relative to the computer 100, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

10 Overview of Components and Operation:

FIG. 2 is a general block diagram illustrating a system for computing and encoding modulated complex lapped transforms in accordance with the present invention. In the system 200 of the present invention, an input signal 206 is received by a sampling device 208, which breaks the signal into blocks. Each block contains L samples, and each new block is formed by discarding the M oldest samples of the block and adding the M newest input samples to the block. In a typical implementation,  $L = 2M$ . Also included in the system 200 are real and imaginary window processors 210, 212 for reducing blocking effects, and real and imaginary transformation processors 220, 222 for coding each block. It should be noted that one window processor with dual real and imaginary computational devices can be used instead of separate real and imaginary window processors. Similarly, one transform processor with dual real and imaginary computational devices can be used instead of separate real and imaginary transform processors.

25 The real and imaginary window processors 210, 212 receive and process the input block by applying and computing butterfly coefficients for the real and imaginary parts of the signal, respectively, to produce resulting real and imaginary vectors. The butterfly coefficients are determined by a given window function, which will be discussed in detail below. The real and imaginary transformation processors 220, 222 compute spatial transforms on the resulting real and imaginary vectors to produce real and imaginary transform coefficients of the MCLT, respectively.

FIG. 3 is a general block/flow diagram illustrating a system and method for computing modulated lapped transforms in accordance with the present invention. In general, the MCLT computation system 300 first receives an input signal 310. Second, a single window processor 312 with real and imaginary computational devices or dual real and imaginary window processors receives a block of M samples of the input signal (box 314). The window processor 312 applies and computes butterfly coefficients, for the real and imaginary parts of the signal (boxes 315, 316), respectively, to produce real and imaginary resulting vectors (boxes 318, 320), respectively.

Third, a single transform processor 322 with real and imaginary computational devices or dual real and imaginary transform processors receives the real and imaginary resulting vectors (box 323). The transform processor 318 performs a discrete cosine transform (DCT) operation on the real vectors (box 324) and a discrete sine transform (DST) operation on the imaginary vectors (box 326). Fourth, real and imaginary output signals are respectively produced as vectors with real and imaginary MCLT coefficients corresponding to the input block of samples (boxes 328, 330). Fifth, the output signal can be processed by transmitting, storing, enhancing, filtering, etc. the signal (box 332). For example, interference within the signal can be reduced with a noise reducer, echo canceller, etc., compression can be achieved by scalar or vector quantization of the MLT coefficients, etc., as desired.

#### Structural and Operational Details of the System

FIG. 4 is a detailed block/flow diagram illustrating a modulated complex lapped transform (MCLT) extended from a modulated lapped transform processor (MLT) in accordance with the present invention. Referring back to FIGS. 2-3 along with FIG. 4, the incoming signal is decomposed into frequency components by a transform processor, such as a modulated complex lapped transform processor (MCLT) of the present invention, that is preferably an extension of a modulated lapped transform processor (MLT). An MLT is preferably the basis for the MCLT because among other things, although other transform processors, such as discrete cosine transforms (DCT and DCT-IV)

are useful tools for frequency-domain signal decomposition, they suffer from blocking artifacts. For example, transform coefficients  $X(k)$  are processed by DCT and DCT-IV transform processors in some desired way, such as quantization, filtering, noise reduction, etc.

5 Reconstructed signal blocks are obtained by applying the inverse transform to such modified coefficients. When such reconstructed signal blocks are pasted together to form the reconstructed signal (e.g. a decoded audio or video signal), there will be discontinuities at the block boundaries. In contrast, the modulated lapped transform (MLT) eliminates such discontinuities  
10 by extending the length of the basis functions to twice the block size, i.e.  $2M$ .

The basis functions of the MLT are obtained by extending the DCT-IV functions and multiplying them by an appropriate window, in the form:

$$a_{nk} = h(n) \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right]$$

where  $k$  varies from 0 to  $M-1$ , but  $n$  now varies from 0 to  $2M-1$ .

15 Thus, the MLT can lead to orthogonal or biorthogonal basis and can achieve short-time decomposition of signals as a superposition of overlapping windowed cosine and sine functions. Such functions provide a more efficient tool for localized frequency decomposition of signals than the DCT or DCT-IV. The MLT is a particular form of a cosine-modulated filter bank that allows for  
20 perfect reconstruction. For example, a signal can be recovered exactly from its MLT coefficients. Also, the MLT does not have blocking artifacts, namely, the MLT provides a reconstructed signal that decays smoothly to zero at its boundaries, avoiding discontinuities along block boundaries. In addition, the MLT has almost optimal performance, in a rate/distortion sense, for transform  
25 coding of a wide variety of signals.

Specifically, the MLT is based on the oddly-stacked time-domain aliasing cancellation (TDAC) filter bank. In general, the standard MLT transformation for a vector containing  $2M$  samples of an input signal  $x(n)$ ,  $n = 0, 1, 2, \dots, 2M-1$  (which are determined by shifting in the latest  $M$  samples of the input signal,  
30 and combining them with the previously acquired  $M$  samples), is transformed

into another vector containing  $M$  coefficients  $X(k)$ ,  $k = 0, 1, 2, \dots, M-1$ . The transformation can be redefined by a standard MLT computation:

$$X(k) \equiv \sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} x(n) h(n) \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right]$$

where  $h(n)$  is the MLT window.

5 Window functions are primarily employed for reducing blocking effects. For example, *Signal Processing with Lapped Transforms*, by H. S. Malvar, Boston: Artech House, 1992, which is herein incorporated by reference, demonstrates obtaining its basis functions by cosine modulation of smooth window operators, in the form:

$$\begin{aligned} p_a(n, k) &= h_a(n) \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \\ p_s(n, k) &= h_s(n) \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \end{aligned} \quad (1)$$

where  $p_a(n, k)$  and  $p_s(n, k)$  are the basis functions for the direct (analysis) and inverse (synthesis) transforms, and  $h_a(n)$  and  $h_s(n)$  are the analysis and synthesis windows, respectively. The time index  $n$  varies from 0 to  $2M-1$  and the frequency index  $k$  varies from 0 to  $M-1$ , where  $M$  is the block size. The MLT is the TDAC for which the windows generate a lapped transform with maximum DC concentration, that is:

$$h_a(n) = h_s(n) = \sin \left[ \left( n + \frac{1}{2} \right) \frac{\pi}{2M} \right] \quad (2)$$

The direct transform matrix  $P_a$  has an entry in the  $n$ -th row and  $k$ -th column of  $p_a(n, k)$ . Similarly, the inverse transform matrix  $P_s$  has entries  $p_s(n, k)$ . For a block  $\mathbf{x}$  of  $2M$  input samples of a signal  $x(n)$ , its corresponding vector  $\mathbf{X}$  of transform coefficients is computed by  $\mathbf{x} = P_a^T \mathbf{X}$ . For a vector  $\mathbf{Y}$  of processed transform coefficients, the reconstructed  $2M$ -sample vector  $\mathbf{y}$  is given by  $\mathbf{y} = P_s \mathbf{Y}$ . Reconstructed  $\mathbf{y}$  vectors are superimposed with  $M$ -sample overlap, generating the reconstructed signal  $y(n)$ .

25 The MLT can be compared with the DCT-IV. For a signal  $u(n)$ , its length- $M$  orthogonal DCT-IV is defined by:



$$U(k) \equiv \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} u(n) \cos \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right]$$

The frequencies of the cosine functions that form the DCT-IV basis are  $(k + 1/2) \pi / M$ , the same as those of the MLT. Therefore, a simple relationship between the two transforms exists. For instance, for a signal  $x(n)$  with MLT coefficients  $X(k)$ , it can be shown that  $X(k) = U(k)$  if  $u(n)$  is related to  $x(n)$ , for  $n = 0, 1, \dots, M/2 - 1$ , by:

$$\begin{aligned} u(n + M/2) &= \Delta_M \{ x(M-1-n) h_a(M-1-n) - x(n) h_a(n) \} \\ u(M/2 - 1 - n) &= x(M-1-n) h_a(n) + x(n) h_a(M-1-n) \end{aligned}$$

where  $\Delta_M \{\cdot\}$  is the  $M$ -sample (one block) delay operator. For illustrative purposes, by combining a DCT-IV with the above, the MLT can be computed from a standard DCT-IV. An inverse MLT can be obtained in a similar way. For example, if  $Y(k) = X(k)$ , i.e., without any modification of the transform coefficients (or subband signals), then cascading the direct and inverse MLT processed signals leads to  $y(n) = x(n - 2M)$ , where  $M$  samples of delay come from the blocking operators and another  $M$  samples come from the internal overlapping operators of the MLT (the  $z^{-M}$  operators).

Assuming symmetrical analysis and synthesis windows, i.e.  $h_a(n) = h_a(2M-1-n)$  and  $h_s(n) = h_s(2M-1-n)$ , it is easy to verify that perfect reconstruction is obtained with:

$$h_a(n) = \frac{h_s(n)}{h_s^2(n) + h_s^2(M-1-n)} \quad (3)$$

Consider the product window  $h_p(n) = h_a(n) h_s(n)$ . From eqn. (3), it follows that:

$$h_p(n) + h_p(n+M) = h_p(n) + h_p(M-1-n) = 1 \quad (4)$$

With either the MLT window in (2) or the biorthogonal windows, the product window satisfies:

$$h_p(n) = \sin^2 \left[ \left( n + \frac{1}{2} \right) \frac{\pi}{2M} \right] = \frac{1}{2} - \frac{1}{2} \cos \left[ \left( n + \frac{1}{2} \right) \frac{\pi}{M} \right] \quad (5)$$

In accordance with the present, a modulated complex lapped transform (MCLT) is derived. The basis functions of the MCLT are defined by cosine and sine modulation of the analysis and synthesis windows, in the form:

$$\begin{aligned} p_a(n, k) &= p_a^c(n, k) - j p_a^s(n, k) \\ p_a^c(n, k) &= h_a(n) \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \\ p_a^s(n, k) &= h_a(n) \sqrt{\frac{2}{M}} \sin \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \end{aligned} \quad (6)$$

5 with  $j \equiv \sqrt{-1}$ , and

$$\begin{aligned} p_s(n, k) &= \frac{1}{2} \left[ p_s^c(n, k) + j p_s^s(n, k) \right] \\ p_s^c(n, k) &= h_s(n) \sqrt{\frac{2}{M}} \cos \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \\ p_s^s(n, k) &= h_s(n) \sqrt{\frac{2}{M}} \sin \left[ \left( n + \frac{M+1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \end{aligned} \quad (7)$$

The MCLT transform coefficients  $X(k)$  are computed from the input signal block  $x(n)$  by  $\mathbf{X} = \mathbf{P}_a^T \mathbf{x}$ , or

$$X(k) = \sum_{n=0}^{2M-1} x(n) p_a(n, k) \quad (8)$$

10 Comparing (1) and (6), it is clear that the MLT of a signal is given by the real part of its MCLT.

Construction of the MCLT can be viewed as providing additional sine-modulated functions as a 2x oversampling in the frequency domain, because for every new  $M$  real-valued input samples the MCLT computes  $M$  complex frequency components. In addition, the MCLT functions above form an over-complete basis. Consequently, the MCLT is in fact a 2x oversampled DFT filter bank (using a doubly-odd DFT instead of the traditional DFT), in which the DFT length size is  $2M$  and the frame (block) size is  $M$ . It should be noted that, unlike in DFT filter banks, the lowest-frequency subband (the "DC" subband) is complex-valued.

20 With the MCLT, if the direct and inverse transforms are cascaded for a block, without modifying the transform coefficients, the following is obtained:

$$\mathbf{X} = \mathbf{P}_a^T \mathbf{x}, \quad \mathbf{Y} = \mathbf{X}, \quad \mathbf{y} = \mathbf{P}_s \mathbf{Y} \Rightarrow \mathbf{y} = \mathbf{P}_s \mathbf{P}_a^T \mathbf{x} \quad (9)$$

with

$$\mathbf{P}_s \mathbf{P}_a^T = \text{diag}\{h_p(n)\} \quad (10)$$

Thus, it should be noted that the mapping from the input block  $\mathbf{x}$  to the reconstructed block  $\mathbf{y}$  is done via a diagonal matrix of order  $2M$ . This is in contrast to the MLT, for which the product  $\mathbf{P}_s \mathbf{P}_a^T$  is not diagonal. In fact, the off-diagonal terms of  $\mathbf{P}_s \mathbf{P}_a^T$  for the MLT are the time-domain aliasing terms, which are cancelled when the overlapped blocks are superimposed. When the subband signals are processed such that  $\mathbf{Y} \neq \mathbf{X}$ , then the time-domain aliasing terms will not cancel exactly, producing artifacts. The MCLT, because of its 2x oversampling, does not rely on time-domain aliasing cancellation.

Moreover, another property of the MCLT is that the reconstruction formula:

$$y(n) = \sum_{k=0}^{M-1} Y(k) p_s(n, k) \quad (11)$$

is achieved. Perfect reconstruction (with  $X(k) = Y(k)$ , of course) can also be achieved with the choices:

$$y_c(n) = \sum_{k=0}^{M-1} \text{Re}\{Y(k)\} p_s^c(n, k) \quad (12)$$

or

$$y_s(n) = \sum_{k=0}^{M-1} \text{Im}\{Y(k)\} p_s^s(n, k) \quad (13)$$

In eqn. (12), an inverse MLT is recognized. Although  $y(n)$ ,  $y_c(n)$ , and  $y_s(n)$  in eqns. (11)–(13) are not block-by-block identical, they build exactly the same reconstructed signal after overlapping.

The magnitude frequency responses of the MCLT filter bank are the same as those of the MLT. For each frequency  $\omega_k = (k + 1/2)\pi/M$  there are two subbands with the same magnitude frequency response but  $\pi/2$  radians out of phase. As such, there is significant overlap among the frequency responses

of neighboring subbands, and the stopband attenuation is around -22 dB with the sine window in eqn. (2).

### Fast Computation

5 As with the MLT, the MCLT can be computed via the type-IV discrete cosine transform (DCT-IV). For a signal  $u(n)$ , its length- $M$  orthogonal DCT-IV is defined by:

$$U(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} u(n) \cos \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \quad (14)$$

10 The frequencies of the cosine functions that form the DCT-IV basis are  $(k + 1/2)\pi/M$ , the same as those of the MLT and MCLT. The type-IV discrete sine transform (DST-IV) of a signal  $v(n)$  is defined by.

$$V(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} v(n) \sin \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \quad (15)$$

For a signal  $x(n)$  with MCLT coefficients  $X(k)$  determined by eqn. (8),  $\text{Re}\{X(k)\} = U(k)$  and  $\text{Im}\{X(k)\} = V(k)$ , if  $u(n)$  in eqn. (14) is related to  $x(n)$ , for

15  $n = 0, 1, \dots, M/2 - 1$ , by:

$$\begin{aligned} u(n + M/2) &= \Delta_M \{x(M-1-n)h_a(M-1-n) - x(n)h_a(n)\} \\ u(M/2-1-n) &= x(M-1-n)h_a(n) + x(n)h_a(M-1-n) \end{aligned}$$

and  $v(n)$  in (15) is related to  $x(n)$  by

$$\begin{aligned} v(n + M/2) &= \Delta_M \{x(M-1-n)h_a(M-1-n) + x(n)h_a(n)\} \\ v(M/2-1-n) &= -x(M-1-n)h_a(n) + x(n)h_a(M-1-n) \end{aligned}$$

where  $\Delta_M\{\cdot\}$  is the  $M$ -sample (one block) delay operator.

20 Thus, the MCLT can be computed from a MCLT computational system 400 having a window processor 410, which receives  $M$  sample blocks 412 of an input signal 414. The window processor 410 has real and imaginary window operators, real and imaginary transform processors 416, 418, such as a length- $M$  DCT-IV and a length- $M$  DST-IV, respectively, as shown in the simplified block diagram of FIG. 4. The real and imaginary window operators of the window

25

processor 410 applies and computes real and imaginary butterfly coefficients, respectively to produce resulting real and imaginary vectors.

After a predefined delay of the real and imaginary vectors, such as a one-block delay, from a real delay block 420 and an imaginary delay block 422, respectively, the length- $M$  DCT-IV 416 receives the real vectors and the length- $M$  DST-IV 418 receives the imaginary vectors. The real transform processor 416 performs a discrete cosine transform (DCT) operation on the real vectors and the imaginary transform processor 418 performs a discrete sine transform (DST) operation on the imaginary vectors. Output signals with real and imaginary parts 424, 426 are produced as vectors with MCLT coefficients corresponding to the input block of samples.

As shown in FIG. 4, for the fast direct MCLT,  $n = 0, 1, \dots, M/2-1$ ,  $k = 0, 1, \dots, M/2-1$ . The DCT-IV and DST-IV can be implemented with the fast techniques. The inverse MCLT can be computed by simply transposing the components, moving the delays to the bottom half outputs of the DCT-IV and DST-IV, replacing the coefficients  $h_a(n)$  by  $h_s(n)$ , and multiplying the contents of the final buffer by  $1/2$ . The fast MCLT computation shown in FIG. 4 does not assume identical analysis and synthesis windows. Therefore, it can be used to compute a biorthogonal MCLT, as long as the windows satisfy the perfect reconstruction condition in eqn. (3).

FIGS. 5A and 5B are detailed diagrams illustrating the window operation of the modulated complex lapped transform of FIG. 4 for the case  $M=8$ . It is easy to infer from those diagrams the general structure for any choice of the block size  $M$ . In general, as shown in FIGS. 5A and 5B, the MCLT computational system 400 of FIG. 4 includes real and imaginary window operators 502, 504. Initially, a first a block of  $M$  samples of an input signal  $x(n)$  is obtained. Second, for each window operator 502, 504 butterfly coefficients 512, 513 are applied and computed to produce resulting real vectors  $u(n)$  and imaginary vectors  $v(n)$ . The butterfly coefficients are determined by a window function  $\{h(n)\}$  514, 515.

For each window operator 502, 504 half of the resulting vectors are stored in a buffer of a one block delay 516, 518 to be used for the next block, while the current contents of the buffer are recovered. Next, the real and

imaginary vectors are received by the real and imaginary transform processors 520, 522, which are preferably discrete cosine transform (DCT) and discrete sine transform (DST) processors to produce vectors with real and imaginary MCLT transform coefficients corresponding to the input signal.

5

Working Operational Example

FIG. 6 is a flow diagram illustrating operational computation of a working example of a modulated complex lapped transform in accordance with the present invention. Referring to FIGS. 3, 4, 5A and 5B along with FIG. 6, first, an input buffer  $x$  containing  $M$  signal samples are read by a MCLT system (box 610).  
10 Second,  $h(n)$  based butterflies are computed for a cosine (real) part  $u(n)$  (box 612), a top half of  $u(n)$  is stored in a buffer 616 for use in the next block (box 614) and the top half of  $u(n)$  is read from a previous block (box 618). Next,  $h(n)$  based butterflies are computed for a sine (imaginary) part  $v(n)$  (box 620), a top half of  
15  $v(n)$  is stored in a buffer 624 for use in the next block (box 622) and the top half of  $v(n)$  is read from a previous block (box 626). A discrete cosine transform, type IV (DCT IV) is then computed on  $u(n)$  (box 628) and a discrete sine transform, type IV (DST IV) is computed on  $v(n)$  (box 630). Last, an output buffer  $U(k)$  containing the real part of the MCLT is produced (box 632) and an output buffer  $V(k)$   
20 containing the imaginary part of the MCLT is produced (box 634).

This allows the present invention to provide perfect reconstruction, in that a signal  $x(n)$  (where  $n$  denotes the discrete-time index) can be recovered exactly from its MCLT coefficients. The MCLT is a linear operator that projects the input block into a frame containing  $2M$  basis functions. The MCLT  
25 corresponds to a tight frame (all blocks of same energy are mapped into transform vectors with the same energy), with a magnitude amplification factor equal to two.

An advantage of the novel MCLT of the present invention is that the MCLT maps a block of  $M$  input signal samples into  $M$  complex frequency  
30 coefficients. As a result, magnitude and phase information at each frequency is immediately available with the MCLT. In addition, the real part of the MCLT is the MLT, which makes for simplified computation of the MLT of a signal whose

MCLT is known. This allows for efficient integration with multimedia applications, such as acoustic echo cancellation and audio coding. Moreover, because the MCLT maps a block of  $M$  input signal samples into  $M$  complex frequency coefficients, the MCLT leads to data expansion of a factor of two. In other words, an oversampling factor of two. This oversampling actually provides good performance in acoustic echo cancellation applications.

In summary, the MCLT of the present invention provides short-time spectral decomposition of signals with explicit magnitude and phase information and perfect signal reconstruction. Fast computability by means of butterflies followed by discrete cosine transform operators. Also, the real of the MCLT can be computed directly by the techniques discussed above, and the imaginary part can be computed with simple modifications. In addition, the MCLT is easily integrated with MLT-based systems. Further, once the MCLT of a signal has been computed, its MLT can be trivially obtained simply by discarding the imaginary part.

#### MCLT Used As An Adaptive Filter

##### **General Overview**

FIG. 7 is a general block diagram of a full-band adaptive filter using an adaptive FIR filtering approach. In general, a filtering system 700, such as a full-band adaptive filtering system, includes an input signal  $x(n)$  710 and reference signal  $r(n)$  712 received by an adaptive filter 720. The adaptive filter produces an output signal  $y(n)$  722, which is sent back into the adaptive filter 720 for providing automatic refinement adjustments to the filtering process until the output signal  $y(n)$  approximates as closely as possible the reference signal  $r(n)$ .

Specifically, the adaptive filter is preferably a filter with time-varying coefficients, which are automatically adjusted such that the output of the filter approximates as closely as possible a prescribed reference signal. If the adaptive filter has a finite impulse response (FIR), the output signal  $y(n)$  is computed from the input signal  $x(n)$  by

$$y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l)$$

where  $L$  is the length of the filter and  $\{w_l(n), l = 0, 1, \dots, L-1\}$  are the time-varying filter coefficients. The adaptive filter allows the output  $y(n)$  to approximate a reference signal  $r(n)$ , or equivalently, drives the error signal  $e(n) = r(n) - y(n)$  as close to zero as possible.

Given an initial setting for the filter coefficient vector  $w_l(0)$  the coefficients can be updated by using a LMS update equation:

$$w_l(n+1) = w_l(n) + 2\mu e(n)x(n-l)$$

where  $\mu$  is a parameter that controls the speed of adaptation. For any coefficient position  $l$ , the LMS performs updates as an adaptation rule if the error  $e(n)$  has the same sign as the input  $x(n)$ , i.e. if their product is positive, then  $|y(n)|$  is too small, and thus  $w_l$  should be increased. The adaptation rule above corresponds to adjusting the coefficient vector  $w$  in the negative direction of the gradient of the error with respect to  $w$ , i.e. a steepest descent update.

The adaptive filter of FIG. 7 can be used in many applications where the response of the filter needs to change in view of varying conditions. Examples include modems (to equalize the signal distortions caused by the telephone lines) and acoustic echo cancellers (to remove the feedback from loudspeakers into microphones). If the input signal has a frequency spectrum that contains many peaks and valleys (i.e., if it is heavily colored), the parameter  $\mu$  in the LMS update equation has to be set to a very low value, which reduces the speed of adaptation, i.e., the speed in which the error signal  $e(n)$  converges to values near zero. On the other hand, if the input signal has a flat (white) spectrum, the LMS update equation is optimal, in the sense that it will lead to the fastest possible convergence.

#### Frequency-Domain Adaptive Filters

FIG. 8 is a general block diagram of a frequency-domain MCLT-based adaptive filter in accordance with the present invention. The performance of the LMS adaptive filter of FIG. 7 discussed above can be improved for colored input signals by using the new structure in FIG. 8. For instance, the signals can be



broken into frequency subbands and an adaptive LMS filter can be performed in each subband, as shown in FIG. 8. Although FIG. 8 depicts the MCLT as the transform operator that performs the frequency decomposition, other transforms could be used, such as a modulated lapped transform (MLT). If real transforms such as the MLT are used, the adaptive filters of each of the subbands have real coefficients. With a complex-valued transform such as the MCLT, the filter coefficients will have complex values.

In general, the frequency-domain adaptive filter of FIG. 8 includes a first MCLT processor 810 for receiving and processing an input signal  $x(n)$  for producing input signal vectors, such as  $X(0)$  through  $X(M-1)$  and a second MCLT processor 812 for receiving and processing a reference signal  $r(n)$  for producing reference signal vectors, such as  $R(0)$  through  $R(M-1)$ . Also included in system 800 are plural adaptive filters 814 for receiving the input signal vectors  $X(0)$  through  $X(M-1)$  and the reference signal vectors  $R(0)$  through  $R(M-1)$  for producing corrected signal vectors, such as  $Y(0)$  through  $Y(M-1)$  and an inverse modulated complex lapped transform processor (IMCLT) 816. The IMCLT 816 receives and processes the corrected signal vectors  $Y(0)$  through  $Y(M-1)$  for producing a final output signal  $y(n)$  that substantially matches the input signal  $x(n)$ .

Thus, in the frequency-domain the adaptive filter of FIG. 8, there is an adaptive filter for each subband  $k$ . Consequently, the subband signals are modified according to the adaptive filter learning computation. The final output  $y(n)$  is obtained by applying an inverse MCLT (IMCLT) on the corrected subband/transform coefficients  $\{Y(k)\}$ . If the original adaptive filter of FIG. 7 had  $L$  coefficients, each adaptive filter in FIG. 8 needs only to have  $L/M$  coefficients, for the same time span.

The advantages of using the frequency-domain adaptive filter of FIG. 8 include faster convergence, because the signals within each subband are approximately white, even for a heavily colored input. Also, the device of FIG. 8 provides improved error control, because the  $\mu$  factors for the adaptive filters in each subband can be adjusted independently. Finally, the system in FIG. 8 can have a reduced computational complexity, because of the fast FFT-based

algorithms available to compute the transforms.

#### **Acoustic echo cancellation (AEC)**

One application of the adaptive filter of FIG. 8 is in acoustic echo  
5 cancellation (AEC), such as for real-time full-duplex communication systems  
(for instance, speakerphones and videoconference systems). For instance, in a  
speakerphone system, the AEC can use an adaptive filter that estimates the  
feedback transfer function from the loudspeaker to the microphone. The  
estimated echo return is then subtracted from the microphone signal. Simple  
10 FIR filters are not ideal because of the length of the impulse response  
necessary to obtain a reasonable amount of echo reduction (for a 16 kHz  
sampling rate and an echo window of 100 ms, a 1,600-point impulse response  
is needed). With subband adaptive filtering, the long FIR full-band filter is  
replaced by a collection of short FIR filters, one for each subband.

15 A critically sampled filter bank such as the MLT can be used for adaptive  
filtering, but the uncanceled aliasing due to subband processing may limit the  
amount of echo reduction to 10 dB or less. Performance can be improved by  
using cross-filters among neighboring subbands, but the extra degrees of  
freedom in such adaptive cross-filters usually slows down convergence  
20 significantly. With the MLCT, subband acoustic echo cancellation (AEC) can  
be performed without cross-filters. Each subband can be processed by a short  
FIR filter with complex taps, as shown in FIG. 8. With a large number of  
subbands, the subband signals are essentially white, and so each adaptive filter  
can be adjusted via the normalized LMS computation.

25 FIG. 9 is a block diagram of a speakerphone working example of the  
adaptive filter of FIG. 8 of the present invention in the form of an acoustic echo  
cancellation device. The speakerphone system 900 of FIG. 9 includes  
communication equipment 910 comprising microphone input signals 912  
received from a microphone 914, which can be amplified by an amplifier 915  
30 and speaker output signals 916 transmitted to a speaker 918. The system 900  
also includes a filter 920, such as the MCLT-based adaptive filter discussed

above, for receiving input signals produced by the speaker 918 and reference signals received by the microphone 914.

For example, in typical speakerphone systems, the local microphone not only captures audio signals intended to be transmitted (such as voice signals of a local person), it also captures audio signals that are being attenuated at the local loudspeaker (such as voice signals from a remote person's transmission) as feedback. Unless the speaker feedback is cancelled, that feedback signal is sent back to the remote person. As such, the remote person will hear an echo of his or her own voice transmitted to the local person.

To solve this problem, the adaptive filter of the present invention includes an MCLT-based adaptive filter for processing and filtering the input and references signal for producing an output signal with information indicating the estimated echo portion of the signal. The estimated echo portion of the output signal is removed or canceled and a resulting clean output signal is sent to the microphone input of the communication equipment 910. Consequently, after an initial audio signal is sent through the system 900, subsequent audio signals with feedback or echoes produced by the loudspeaker 918 are canceled by the adaptive filter 920 before the microphone input is received

Since the input to the adaptive filter 920 is the signal from the speaker 918 and the reference input is the signal from the microphone 914, the output of the adaptive filter will be a good estimate of the portion of the microphone signal that is dependent on the loudspeaker signal, which is precisely the echo. When the echo is subtracted from the signal of the microphone 914, as shown in FIG. 9, only the part of the microphone signal that is not correlated with the loudspeaker signal will remain. The remaining part (which is the "cleaned" microphone signal 912 in FIG. 9) corresponds to the other local sounds, such as the voice of the person speaking and other ambient sounds.

FIG. 10 is a general block diagram of an acoustic echo cancellation device (AEC) with MCLT-based adaptive filters in accordance with the present invention. In general, referring to FIGS. 8-9 along with FIG. 10, the AEC 1000 of FIG. 10 includes a first MCLT processor 1010 for receiving and processing a loudspeaker signal as an input signal  $x(n)$  for producing input signal vectors

$X(0)$  through  $X(M-1)$  and a second MCLT processor 1012 for receiving and processing a microphone signal as a reference signal  $r(n)$  for producing reference signal vectors  $R(0)$  through  $R(M-1)$ .

Also included in system 1000 are plural adaptive filters 1014 for  
5 receiving the input signal vectors  $X(0)$  through  $X(M-1)$  and the reference signal vectors  $R(0)$  through  $R(M-1)$ . The adaptive filters estimate the echo within the signals, which are then combined with the reference signals for canceling the echoes and producing cleaned and corrected signal vectors, such as  $Y(0)$  through  $Y(M-1)$ . An inverse modulated complex lapped transform processor  
10 (IMCLT) 1016 receives and processes the corrected signal vectors  $Y(0)$  through  $Y(M-1)$ , which have MCLT coefficients without echo, for producing a final output signal with the echo canceled.

Subtraction of the estimated echo from the microphone signal is preferably performed for each subband, resulting in a set of subband signals  
15  $Y(k)$  with the echo substantially removed.

The AEC and spectral subtraction can be combined using a single MLCT decomposition. For example, spectral subtraction can be applied to the subband signals immediately after the AEC adaptive filters. If the resulting signal is to be encoded by an MLT-based codec, then the MLT coefficients for  
20 the audio codec can be obtained by simply taking the real part of the outputs of the spectral subtraction. Therefore, only a single transformation step with the MCLT is necessary to perform simultaneous signal enhancement and coding.

If the waveform  $y(n)$  corresponding to the echo-cancelled subband signals  $Y(k)$  in FIG. 10 is desired, then an inverse modulated complex transform  
25 (IMCLT) can be performed on  $Y(k)$ , as shown in FIG. 8. However, if the signals are to be encoded with an MLT-based coder/decoder (codec), such as MSAudio, then  $y(n)$  need not be computed since a codec can work directly with the  $Y(k)$  subband/transform coefficients.

One advantage of using the MCLT-based adaptive filters is that the  
30 MCLT uses short windows, which leads to low processing delay. Another advantage is that the MCLT allows for perfect signal reconstruction. Also, integrating an MCLT adaptive filter with an MLT-based processing system (for

example, an audio codec) is very easy, since the MLT is obtained directly as the real part of the MCLT. Further, for a given number of subbands  $M$  (which is also the block size), a windowed Fourier transform decomposes the signal into  $M/2+1$  distinct subbands. The MCLT breaks the signal into  $M$  subbands, and so it provides essentially twice the frequency resolution. Therefore, an MCLT-based adaptive filter will converge faster, because narrower subbands tend to have a flatter spectrum.

FIG. 11 is a wave signal illustrating sample results of the working example of FIG. 9. The first wave signal (Wave A) is the microphone signal as a recorded echo return. The second wave signal (Wave B) is the output of a full-band AEC, echo reduction ratio (ERR)  $\approx 26$  dB (it should be noted that the vertical scale is different). The third wave signal (Wave C) is an output of a 512-band MLT AEC without cross filters, ERR  $\approx 5$  dB. The bottom wave signal (Wave D) is an output of a 512-band MCLT AEC without cross filters, ERR  $\approx 20$  dB.

Specifically, the original signal is an actual echo return recorded at 16 kHz sampling from a microphone located at about 20" from the loudspeaker (using a 4" driver). The signals in FIG. 11 show the cancelled echo after convergence of each AEC (which takes a few seconds in all cases). The MLT and MCLT AECs used  $M = 512$  subbands and a four-tap adaptive filter in each band (corresponding to an echo window of about 128 ms). The echo attenuation for the MCLT is about 20 dB, which is adequate for many practical teleconferencing applications.

## 25 Noise Reduction

In addition, the MCLT of the present invention is amenable to other types of frequency-domain processing while allowing for perfect signal reconstruction. For instance, another kind of processing that can be efficiently performed in the frequency domain, especially with the MCLT, is noise reduction. For the audio/voice communication system in FIG. 9, even after the loudspeaker echo is cancelled the signal may still be noisy. The AEC usually removes feedback from the loudspeaker, but may not remove other noises, such as ambient

noises that may be generated by computers and fans in an office.

An efficient approach to attack noise reduction is with spectral subtraction. For each subband  $k$ , the signal  $Y(k)$  is considered as having a desired signal and a noise component, in the form:

5

$$Y(k) = S(k) + N(k)$$

where  $S(k)$  is the desired signal and  $N(k)$  is the interfering noise. Assuming the signal and noise are uncorrelated, the energy of the subband signal is just the sum of the signal and noise energies:

10

$$|Y(k)|^2 = |S(k)|^2 + |N(k)|^2$$

With spectral subtraction, noise reduction is achieved by estimating the average noise magnitude  $|N(k)|$  during low-amplitude signals, i.e., during periods where  $|S(k)|$  is assumed to be zero. The variable  $N_e(k)$  is the noise level estimate for the  $k$ th subband, which can be subtracted from  $Y(k)$ , in the form

15

$$|Y_f(k)| = |Y(k) - \alpha(k)|N_e(k)|, \quad (A)$$

20

where  $Y_f(k)$  is the filtered signal. As such, a portion of the estimated magnitude noise is subtracted from the magnitude of each subband signal. The phase is not affected, since the average noise phase is always zero. The parameters  $\alpha(k)$  control how much of the noise estimate is subtracted from each subband signal, and so  $0 < \alpha(k) < 1$  is preferably set. These parameters are preferably adjusted depending on the quality of the noise estimates. For example, if the noise estimate is significantly above the true noise level, the subtraction in eqn. (A) will remove part of the signal, also, leading to noticeable artifacts.

25

Specifically, FIG. 12 is a general block diagram of a noise reduction device with MCLT-based adaptive filters in accordance with the present invention. In general, the noise reduction device 1200 of FIG. 12 includes an MCLT

30

FIG. 13 is a flow diagram illustrating operational computation of a noise reduction device incorporating the modulated complex lapped transform of the present invention. Referring to FIG. 12 along with FIG. 13, first, an input buffer x containing M subband coefficients are read (box 1310) by MCLT processor 1210 of FIG. 12. Second, the MCLT coefficients  $X(k)$  are computed (box 1312) and this information is sent to the subband subtraction device 1212 of FIG. 12. Third, a first subband  $k=0$  is analyzed (box 1314) by the subband subtraction device 1212 of FIG. 12. Fourth, it is determined whether a transform coefficient  $|X(k)|$  is less than a threshold value,  $Th$ . If it is, the noise level estimate is adjusted (box 1318), for example with an update function such as:

If the coefficient  $|X(k)|$  is not less than the threshold value,  $Th$ , and after the

above function is performed, spectral subtraction is performed (box 1320) by the subband subtraction device 1212 of FIG. 12, preferably with the following expression:

$$|Y_f(k)| = |X(k)| - \alpha(k)|N_e(k)|$$

5

Next, the subband subtraction device processes the next subband  $k=k+1$  (box 1322). It is then determined whether  $k=M$  (box 1324). If not, the process returns to step 1316. Otherwise, last, an output buffer  $Y(k)$  containing  $M$  filtered subband coefficients is produced (box 1326) with reduced noise by the IMCLT 1216 of

10

FIG. 12. In practice, the noise reduction process is preferably performed right after the echo cancellation process of FIG. 10, otherwise the loudspeaker echo would interfere with the noise estimate and make it less reliable.

Although the spectral subtraction as in eqn. (A) can be performed with  
15 subband signals derived from a windowed Fourier transform, there are several advantages of using an MCLT instead of the Fourier transform. First, with the MCLT, perfect reconstruction of the signal can be obtained, which is important in low-noise, high fidelity applications. However, with a windowed Fourier  
20 transform, usually long windows are needed for good enough signal reconstruction, increasing the processing delay. Next, as discussed above, for a given number of subbands  $M$  (which is also the block size), a windowed Fourier transform decomposes the signal into  $M/2+1$  distinct subbands. In contrast, the MCLT breaks the signal into  $M$  subbands, and so it provides essentially twice the frequency resolution. Therefore, an MCLT-based noise  
25 reducer allows for finer discrimination, which is important in reducing noise with periodic components, such as high-frequency tones generated by computer hard disks.



FIG. 14 is a wave signal illustrating sample results of the working example of FIGS. 12-13 using the MCLT of the present invention with spectral subtraction. The top wave signal (Wave A) is the original speech, corrupted by PC noise, SNR  $\approx$  15 dB and the bottom wave signal (Wave B) is the processed speech, SNR  $\approx$  30 dB.

Specifically, an original 8-second speech signal was captured at 16 kHz sampling rate, with the microphone near a very noisy personal computer (PC), whose noise spectrum is approximately pink. The depth of subtraction for a noise reduction of about 15 dB was adjusted. The results are shown in FIG. 14, where the signal-to-noise ratio (SNR) was successfully increased from 15 dB to 30 dB. More importantly, the processed file has fewer artifacts than the results obtained using a commercial product that uses standard DFT filter banks for spectral subtraction.

#### 15 ***Integrating AEC, Noise Reduction and Codec***

FIG. 15 is a block diagram of a working system of the present invention shown as a signal enhancer and noise reducer integrated with a codec. The MCLT, AEC and noise reducer of the present invention as described above, can be integrated with an audio codec for use by a real time communication system, such as audio applications including Internet telephony or other forms of hands-free teleconferencing or telephony. The integration of the MCLT, AEC, noise reducer and codec leads to improved performance, reduced computational complexity, and reduced memory usage and processing delay over current systems.

In general, referring to FIG. 15, the AEC (element 800 of FIG. 8) and noise reducer (element 1200 of FIG. 12) of the present invention can be integrated with an audio codec to form a novel integrated processor 1500. The integrated processor 1500 of the present invention includes a MCLT processor 1510 (similar to MCLT processor 810 of FIG. 8), an AEC processor 1512 (similar to the AEC processor 800 of FIG. 8), a noise reducer 1516 (similar to the noise reducer 1200 of FIG. 12), a coefficient filter 1516, a magnitude processor 1518, and a codec 1520, which can be any suitable codec. The

noise reducer 1516 is preferably included in the codec 1520, as shown in FIG. 15. The codec can be the audio codec (with suitable modifications in accordance with the present invention) described in co-pending U.S. Patent Application Serial No. 09/085,620, filed on May 27, 1998 by Henrique Malvar, entitled "Scalable Audio Coder and Decoder" and assigned to the current assignee, which is herein incorporated by reference.

During operation, in a real-time communication system (i.e. an application that utilizes a digital communication channel), such as a digital network or online Internet communications for dynamically transmitting audio and video signals, the audio that is captured can be enhanced by operations of the AEC 1512 and noise reducer 1514 of the present invention. After enhancement, the audio signal is preferably coded (compressed) by the codec 1520 to limit the bit rate to a rate that is adequate for the communication channel.

As shown in FIG. 15, the MCLT processor 1510 receives and processes an input signal  $x(n)$  for producing input signal vectors, such as  $X(0)$  through  $X(M-1)$  and the AEC processor 1512 receives and processes a reference signal for producing reference signal vectors 1522. The noise reducer 1516 then receives the signal vectors 1514 from the AEC processor 1512 and produces enhanced MCLT coefficients (real and imaginary parts), such as coefficients  $Y(0)$  through  $Y(M-1)$ , in accordance with the noise reducer 1200 of FIG. 12.

Specifically, in a communication system, the AEC does not need to perform an inverse MCLT after the adaptive filters in each subband, if a codec operates in the frequency domain. Thus, the same principle applies when the noise reduction process of eqn. (A) is added. For example, after computing the subband signals  $Y(k)$  at the output of the AEC, the noise reduction step is applied to generate the filtered subband coefficients  $Y(k)$ , which are then sent directly to the codec, without the need to return to the corresponding time-domain signal. Alternatively, if the codec does not operate in the frequency domain (such as many telephony codecs), then the echo-cancelled and noise filtered signal  $y(n)$  can be obtained simply by computing an inverse MCLT on the subband signals  $Y(k)$ .

It should be noted that instead of using the modulated lapped transform (MLT) as the first processing step (the audio coder of co-pending U.S. Patent Application Serial No. 09/085,620), the present invention uses the MCLT as the first processing step in order to avoid performing inverse MCLT computations. This is because, as described above, the MLT is the real part of the MCLT. Thus, if the enhanced audio signal is available in the MCLT domain, it is not necessary to compute the inverse MCLT to recover the time domain waveform and then compute its MLT. Instead, the imaginary part of the MCLT coefficients are discarded, thereby allowing the system 1500 to obtain the MLT coefficients directly from the real part of the MCLT.

From the  $Y(k)$  enhanced MCLT coefficients produced by the AEC 1512 and noise reducer 1514, coefficients  $R(k)$  can be obtained by:

$$R(k) = \text{Re}\{Y(k)\}$$

where  $\text{Re}\{x\}$  denotes taking the real part. In particular, as shown in FIG. 15, the coefficient filter 1516 and the magnitude processor 1518 receive coefficients  $Y(0)$  through  $Y(M-1)$ . The coefficient filter 1516 processes the MCLT coefficients (real and imaginary parts) and discards the imaginary parts of the MCLT coefficients.

#### ***Improving Computation of the Auditory Masking Functions***

In addition to obtaining the MLT coefficients needed by the codec directly from the real part of the MCLT coefficients, the integrated system 1500 of FIG. 15 produces accurate masking functions. For example, the audio codec of U.S. Patent Application Serial No. 09/085,620 computes weighting functions based on hearing thresholds, defined by functions that approximate the masking phenomena in the human auditory system. Such masking functions can be computed based on the power spectrum of the incoming audio, i.e. the power values at each frequency index  $k$ .

As such, the spectral magnitudes are approximated by magnitudes of the MLT coefficients.

However, since the MLT coefficients are obtained from projected the signal into modulated cosines, their magnitudes are typically not directly proportional to the actual physical r.m.s. (root mean-square) power contained in the signal at each frequency subband. With the MCLT, the magnitudes can be computed directly from the real and imaginary parts (cosine and sine projections, respectively), and such magnitudes are then directly proportional to the physical r.m.s. power at each frequency subband  $k$ . In that way, the computation of the masking functions are more precise as compared to computations based solely on the MLT (real part) coefficients.

As shown in FIG. 15, the magnitude processor 1518 computes the magnitudes of the MCLT coefficients, such as  $U(0)$  through  $U(M-1)$ . Computation of the magnitudes  $U(k)$  can be performed by the following expression:

$$U(k) = \sqrt{\text{Re}\{Y(k)\}^2 + \text{Im}\{Y(k)\}^2}$$

The codec 1520 further includes a weighting processor 1524, a masking functions processor 1526 and an encoding processor 1528. The masking functions processor 1526 receives the magnitude coefficients produced by the magnitude processor 1518 and computes masking functions. The weighting processor 1524 receives the masking functions and the real part of the signal from the coefficient filter 1516, such as  $R(0)$  through  $R(M-1)$  for producing the weighted signal, as described above. Last, the encoding processor 1528 performs quantization and encoding processing to produce the output bitstream.

As shown in FIG. 15, the input signal is transformed from the time domain to the frequency domain once, by means of an MCLT. Therefore, the enhancement functions of the AEC 1512 and the noise reducer 1514 are computed on the MCLT coefficients, while the codec 1520 uses the real part of the MCLT coefficients for quantization and encoding and the magnitudes of the

MCLT coefficients for computation of precise auditory masking functions and weighting functions.

#### Integrating AEC, Noise Reduction, Coding, and Speech Recognition

5 For speech recognition applications, the computational load can be minimized by performing the MCLT computation once by integrating several components that process the signal in the frequency domain. As shown in FIG. 15, a speech recognizer 1530 having a coefficient processor 1532 and recognition engine 1534. The speech recognizer 1530 can be located after the  
10 magnitudes are computed 1518.

In general, in most automatic speech recognition (ASR) systems, the incoming speech signal is divided into blocks of 10 to 30 ms duration. For each block, a cepstrum vector is computed, and cepstral coefficients are used for the next step of statistical and language pattern analysis. With the set of Fourier  
15 transform coefficients for the input signal block  $\{X(k)\}$ , the cepstral coefficients  $V(r)$  can be defined by:

$$V(r) = \sum_{k=0}^{M-1} \log|Y(k)| e^{j \frac{2\pi rk}{M}}, \quad r = 0, 1, \dots, N$$

20 which is the inverse Fourier transform of the log magnitude spectrum of the block. The parameter  $N$  (the number of spectral coefficients computed) can be set between 10 and 20.

To compute the Fourier transform coefficients  $X(k)$ , a fast Fourier transform (FFT) operator is preferably computed on the incoming block. When  
25 the incoming signal has already been processed by an MCLT-based AEC and noise reducer, however, a spectral representation of the signal is already computed, namely, the MCLT coefficients.

Thus, an approximate cepstral vector  $C(r)$  can be computed by the coefficient processor 1532 using the MCLT coefficients instead of other  
30 coefficients (such as FFT coefficients), such that:

$$C(r) = \sum_{k=0}^{M-1} \log|U(k)| e^{j \frac{2\pi r k}{M}}, \quad r = 0, 1, \dots, N$$

where  $\{U(k)\}$  is the set of MCLT coefficients. The speech recognition engine 1534 receives these coefficients for performing speech recognition.

5 Although this new cepstral vector  $C(r)$  is not identical to the original cepstral vector  $V(r)$ , the patterns present in  $V(r)$  will also be present in  $C(r)$ . Re-training of the spectral recognition engine can be performed, so it will re-adapt to the typical patterns in  $C(r)$ .

The foregoing description of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

15

## WHAT IS CLAIMED IS:

1. A computer-implemented method (1000) for processing an audio signal, the method comprising:

5 applying butterfly coefficients determined by a real and an imaginary window function to a received input signal to respectively produce real and imaginary resulting vectors (1010, 1012);

10 computing real and imaginary spatial transforms of the real and imaginary resulting vectors, respectively, to produce a modulated complex lapped transform having real and imaginary transform coefficients as an encoded output (1010, 1012); and

15 producing enhanced complex frequency coefficients from the transform coefficients (1014) and discarding the imaginary portions of the enhanced complex frequency coefficients to produce filtered real coefficients (1016).

2. The method of claim 1, wherein the encoded output is produced as a vector with modulated complex lapped transform coefficients corresponding to the input signal (328, 330).

20 3. The method of claim 1, further comprising processing the encoded output (332) by at least one of transmitting the output, storing the output, compressing the output, enhancing the output, and filtering the output.

25 4. The method of claim 1, wherein producing enhanced complex frequency coefficients comprises is performed with an acoustic echo canceller (900).

30 5. The method of claim 3, wherein filtering the output comprises reducing interference within the input signal with a cancellation device (900).

6. The method of claim 3, wherein compressing the output (332) is achieved by at least one of scalar and vector quantization.

7. The method of claim 1, wherein the window functions (315, 316) are adapted for reducing blocking effects.

5 8. The method of claim 1, wherein the real spatial transform is performed by a discrete cosine transform operation (324).

9. The method of claim 1, wherein the imaginary spatial transform is performed by a discrete sine transform operation (326).

10

10. The method of claim 1, wherein half of the resulting vectors are stored in a memory of a one block delay buffer (420, 422).

11. An audio processor (900), comprising:  
15 a modulated complex lapped transform processor receiving and spectrally decomposing input signals (914, 918) into complex frequency coefficients having real and imaginary portions (810, 812) associated with the input signals (914, 918);  
an enhancer comprising an acoustic echo cancellation processor  
20 and a noise reducer (900), wherein the acoustic echo cancellation processor and the noise reducer receive the complex frequency coefficients and produce enhanced complex frequency coefficients (920); and  
an encoder device (920) receiving the enhanced complex  
coefficients and encoding them.

25

12. The audio processor of claim 11, further comprising producing an output signal (916) as a vector with the transform coefficients corresponding to a portion of the input signals.

30 13. The audio processor of claim 12, further comprising an external module (332) for processing the output signal by at least one of transmitting the output, storing the output, compressing the output, enhancing the output, and



filtering the output.

14. The audio processor of claim 11, wherein the modulated lapped transform processor includes a window processor (312) capable of applying butterfly coefficients determined by a real and an imaginary window function to  
5 respectively produce real and imaginary resulting vectors (318, 320).

15. The audio processor of claim 11, wherein the window functions (312) are adapted for reducing blocking effects.

10

16. The audio processor of claim 11, wherein the modulated lapped transform processor has a real transform module with a discrete cosine transform operator (324) and an imaginary transform module with a discrete sine transform operator (326).

15

17. The audio processor of claim 12, wherein the output signal is produced as a vector with biorthogonal modulated complex lapped transform coefficients (916) corresponding to the input signal.

20

18. The audio processor of claim 14, wherein the window processor (940) further comprises a memory of a one block delay buffer (420, 422) for storing a portion of the respective resulting vectors in the memory and for recovering current contents of the delay buffer.

25

19. The audio processor of claim 11, wherein the encoder device (940) encodes the real part of the complex frequency coefficients.

30

20. The audio processor of claim 11, wherein the encoder device (940) computes quantization weighting functions from the magnitudes of the modulated complex lapped transform coefficients.

21. A method for recognizing human speech (1500), comprising:

applying butterfly coefficients determined by a real and an imaginary window function to a received input signal to respectively produce real and imaginary resulting vectors (1510);

5 computing real and an imaginary spatial transforms of the real and imaginary resulting vectors, respectively, to produce a modulated complex lapped transform having real and imaginary transform coefficients as an encoded output (1510);

10 producing enhanced complex frequency coefficients from the transform coefficients and discarding the imaginary portions of the enhanced complex frequency coefficients to produce filtered real coefficients (1520); and

computing an approximate cepstral vector from the enhanced complex frequency coefficients and the filtered real coefficients (1530).

22. The method of claim 19, further comprising performing human  
15 speech recognition from the approximate cepstral vector (1530).

1/13

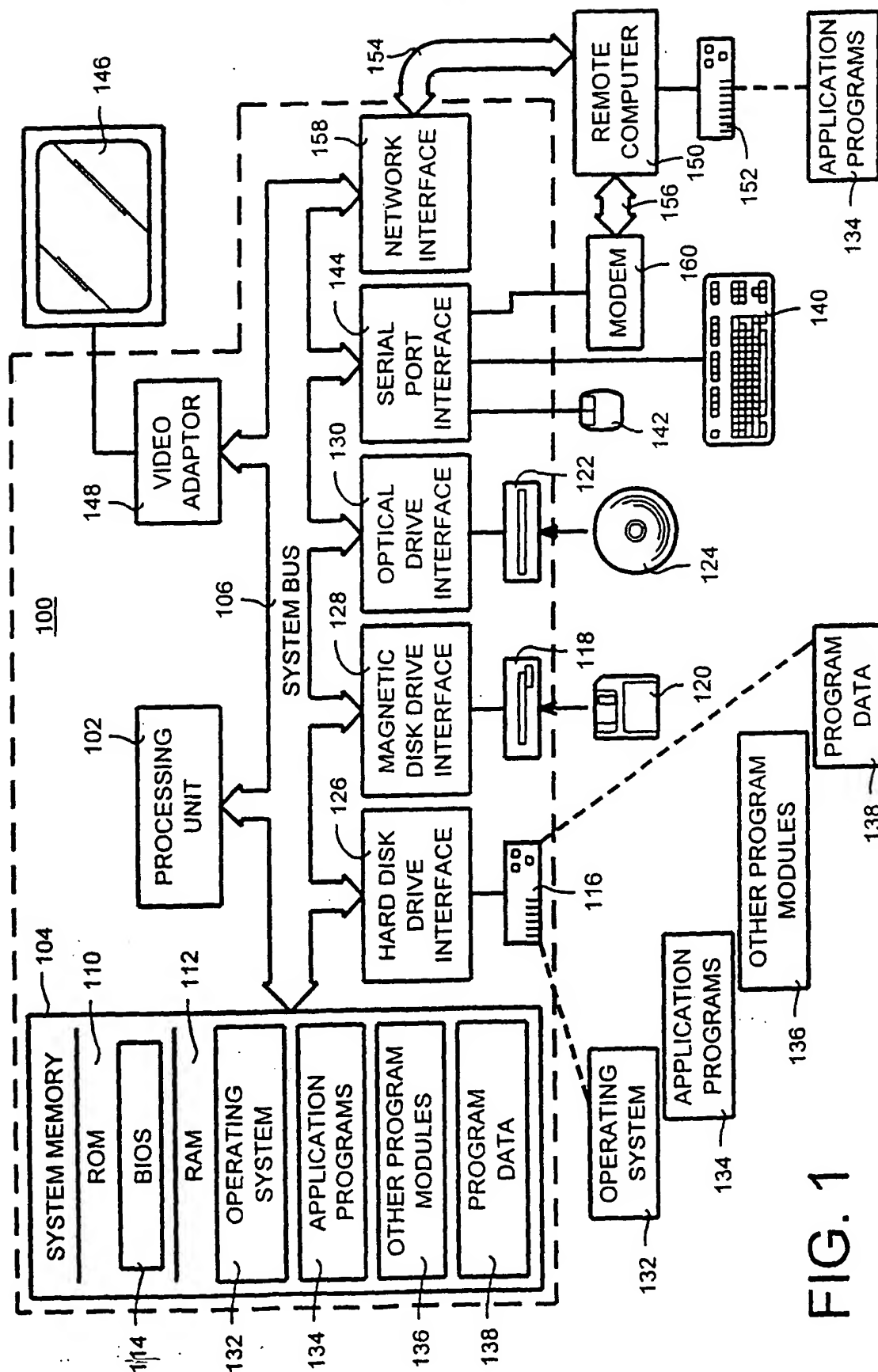
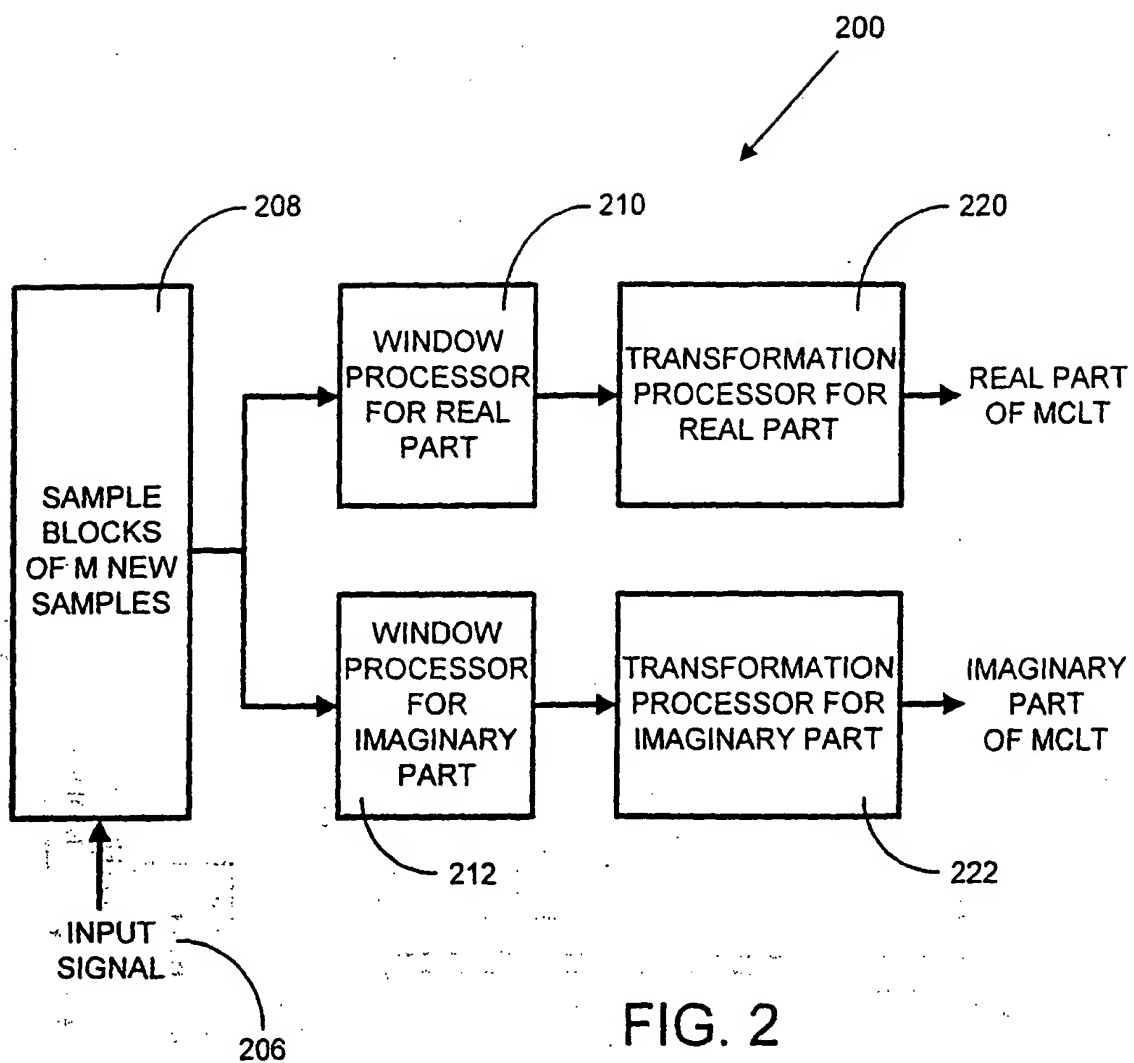


FIG. 1

2/13



3/13

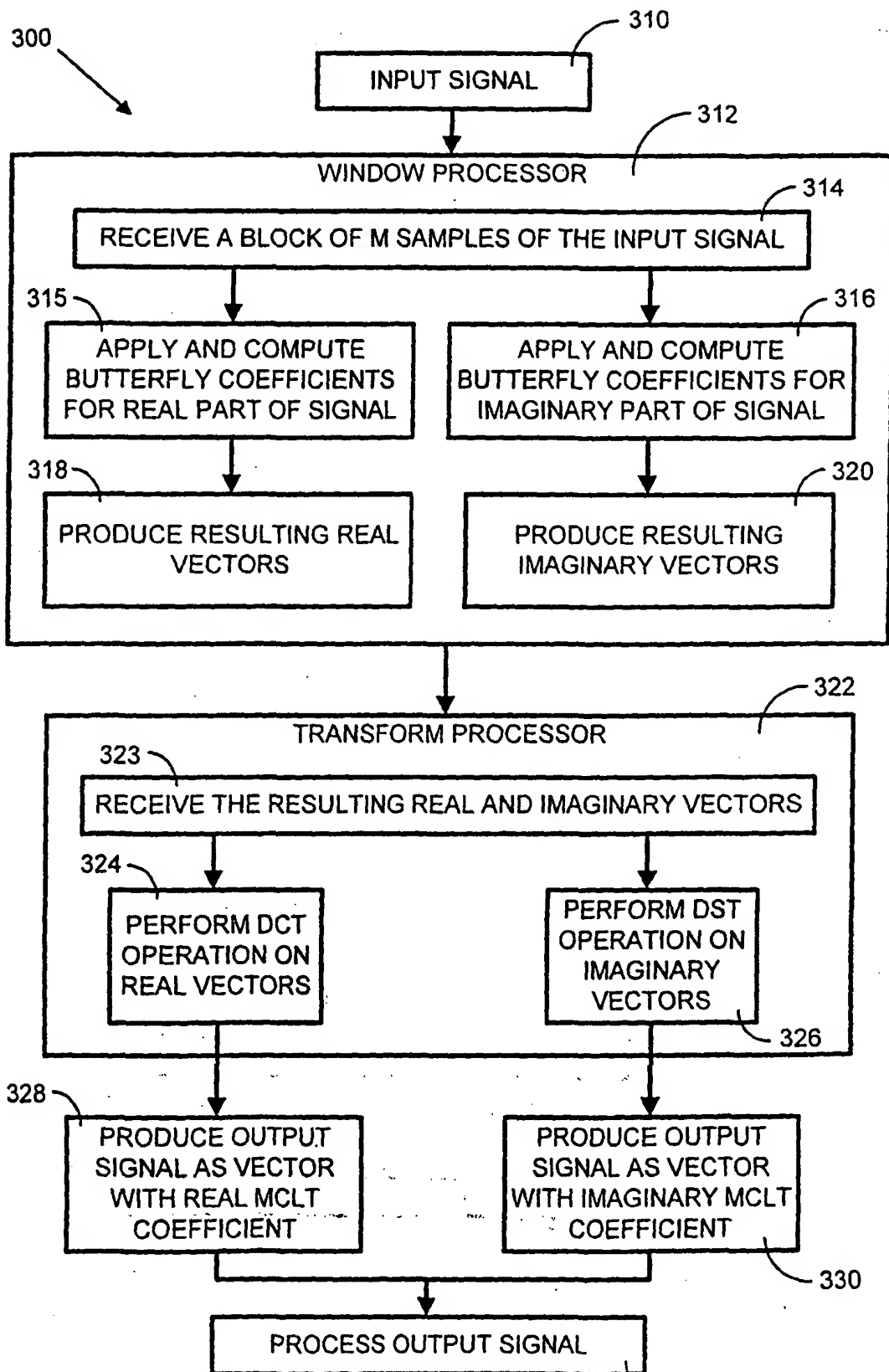


FIG. 3

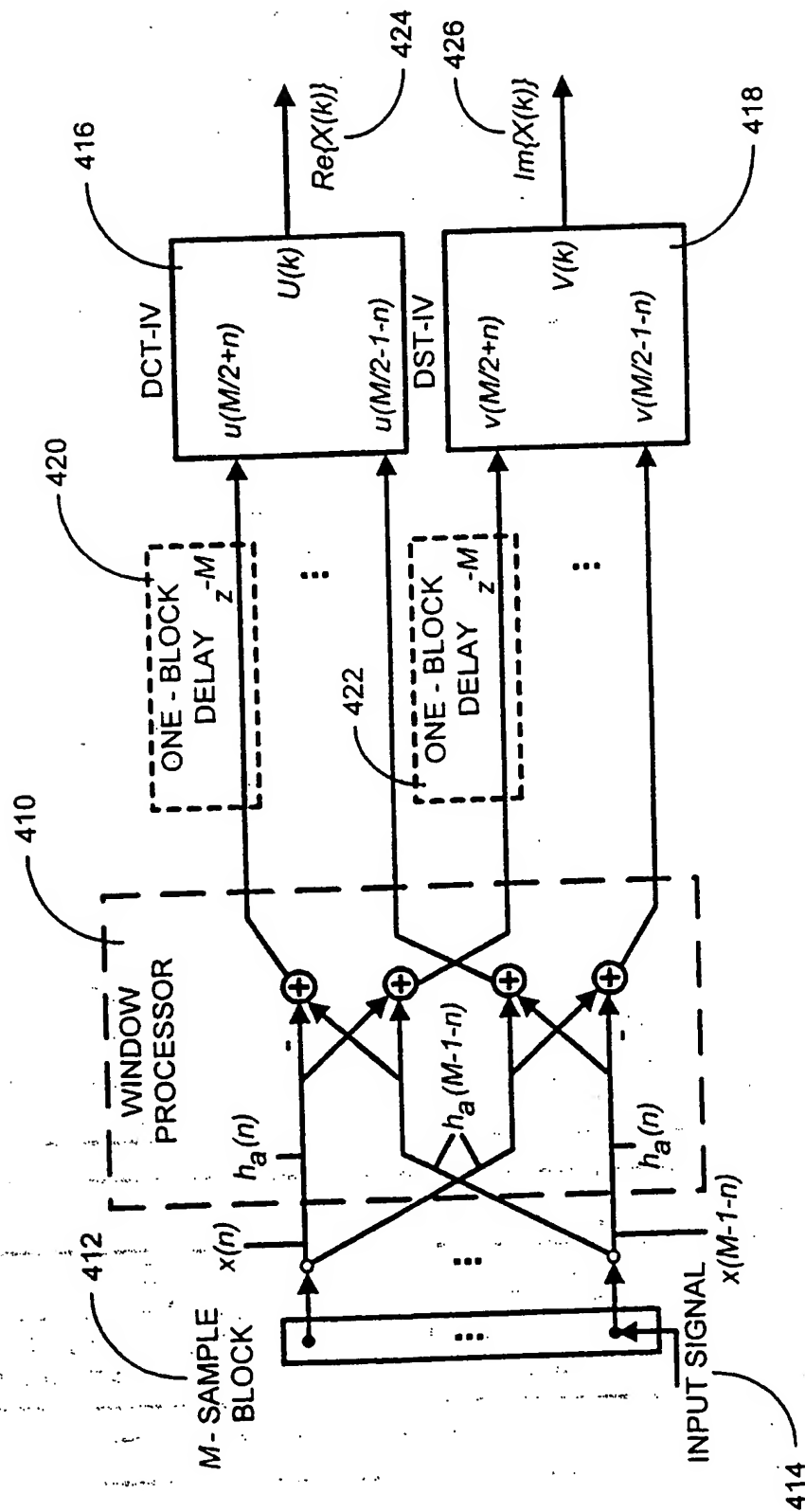


FIG. 4

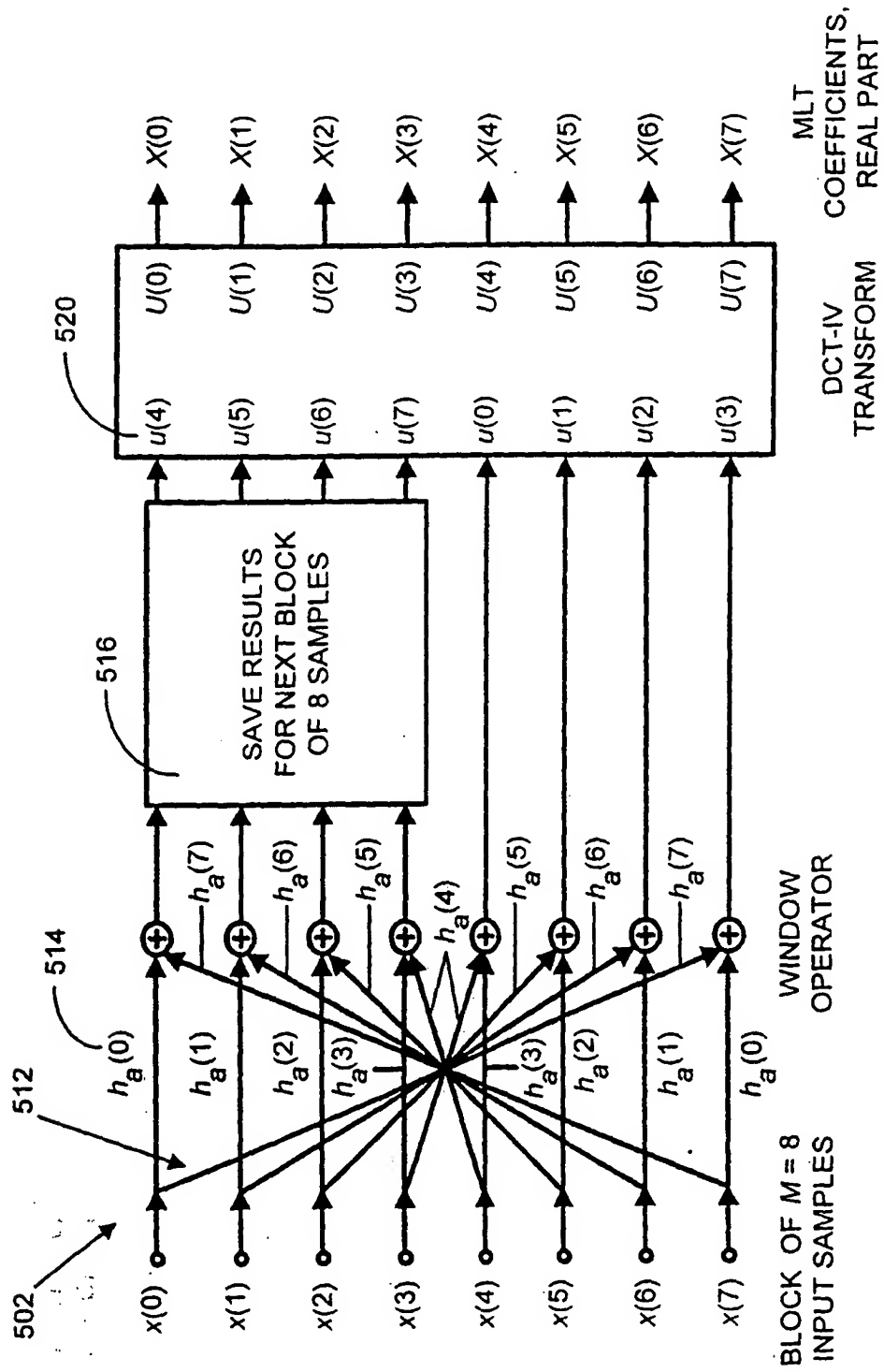


FIG. 5A

6/13

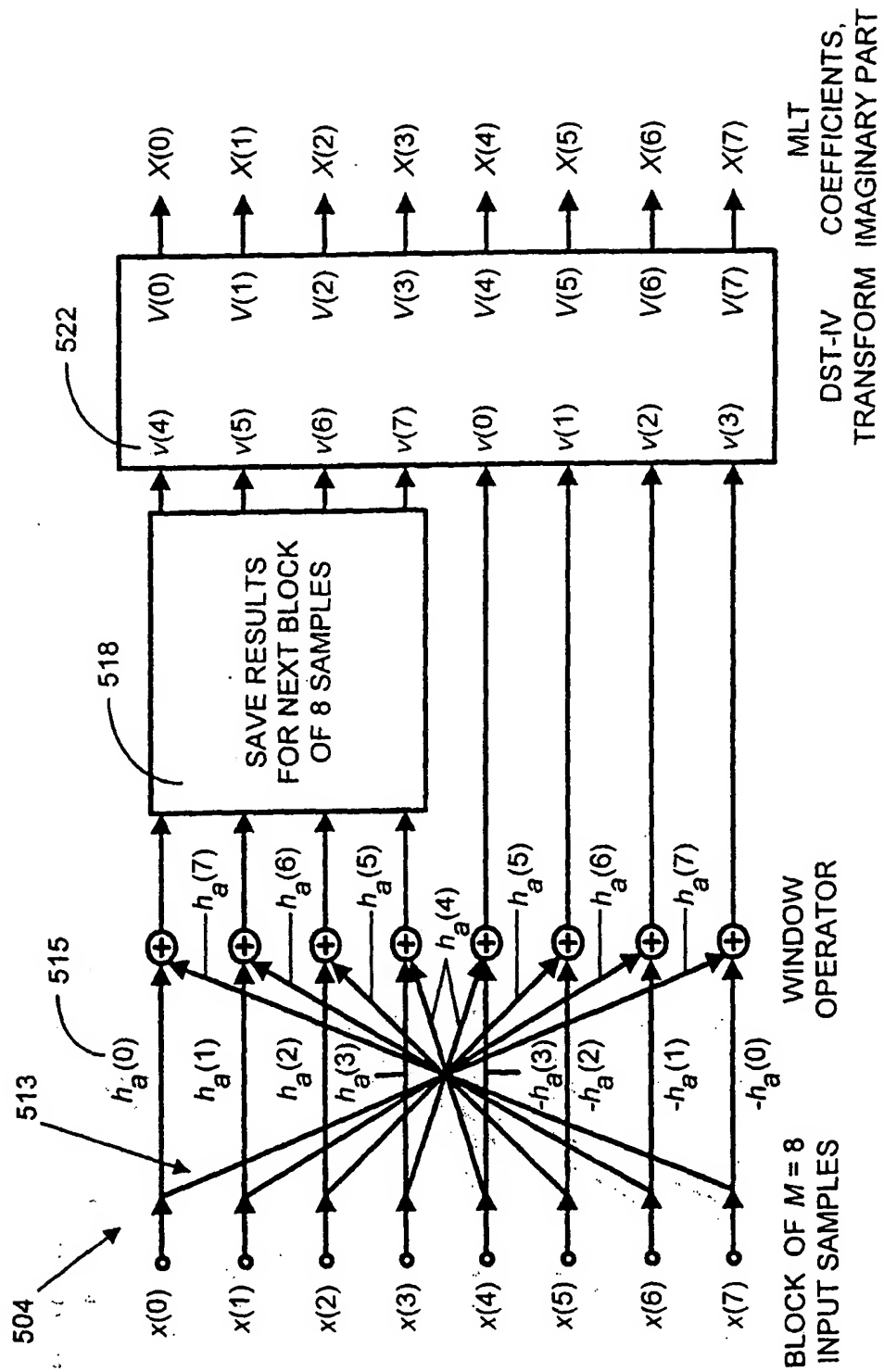


FIG. 5B



7/13

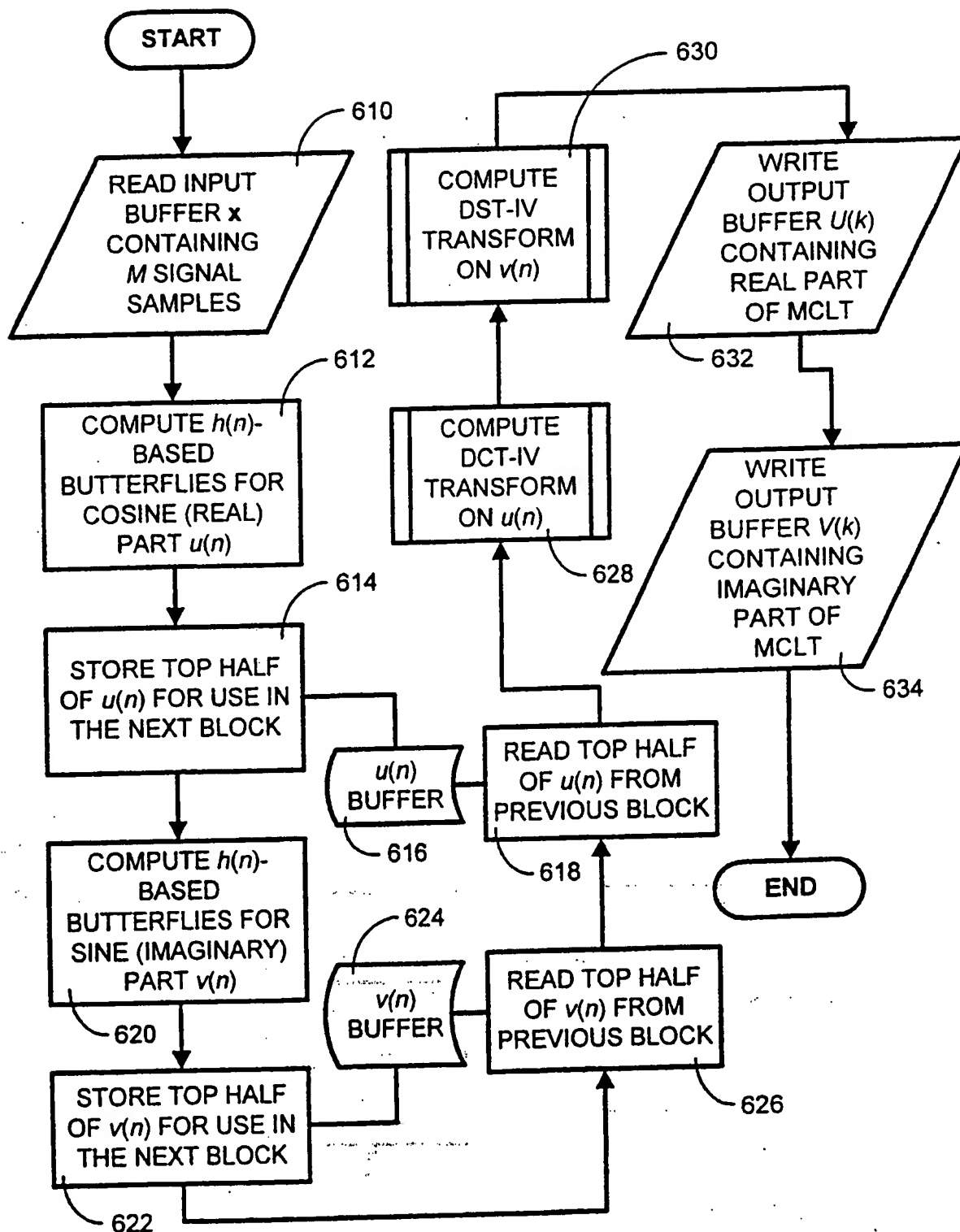


FIG. 6

8/13

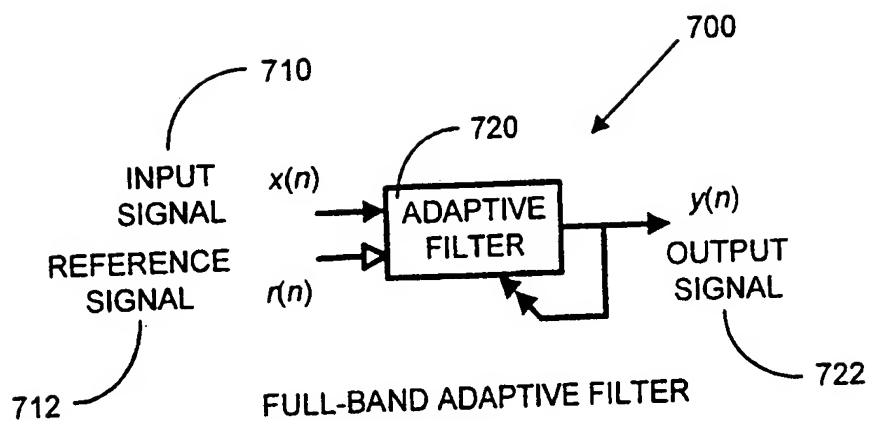
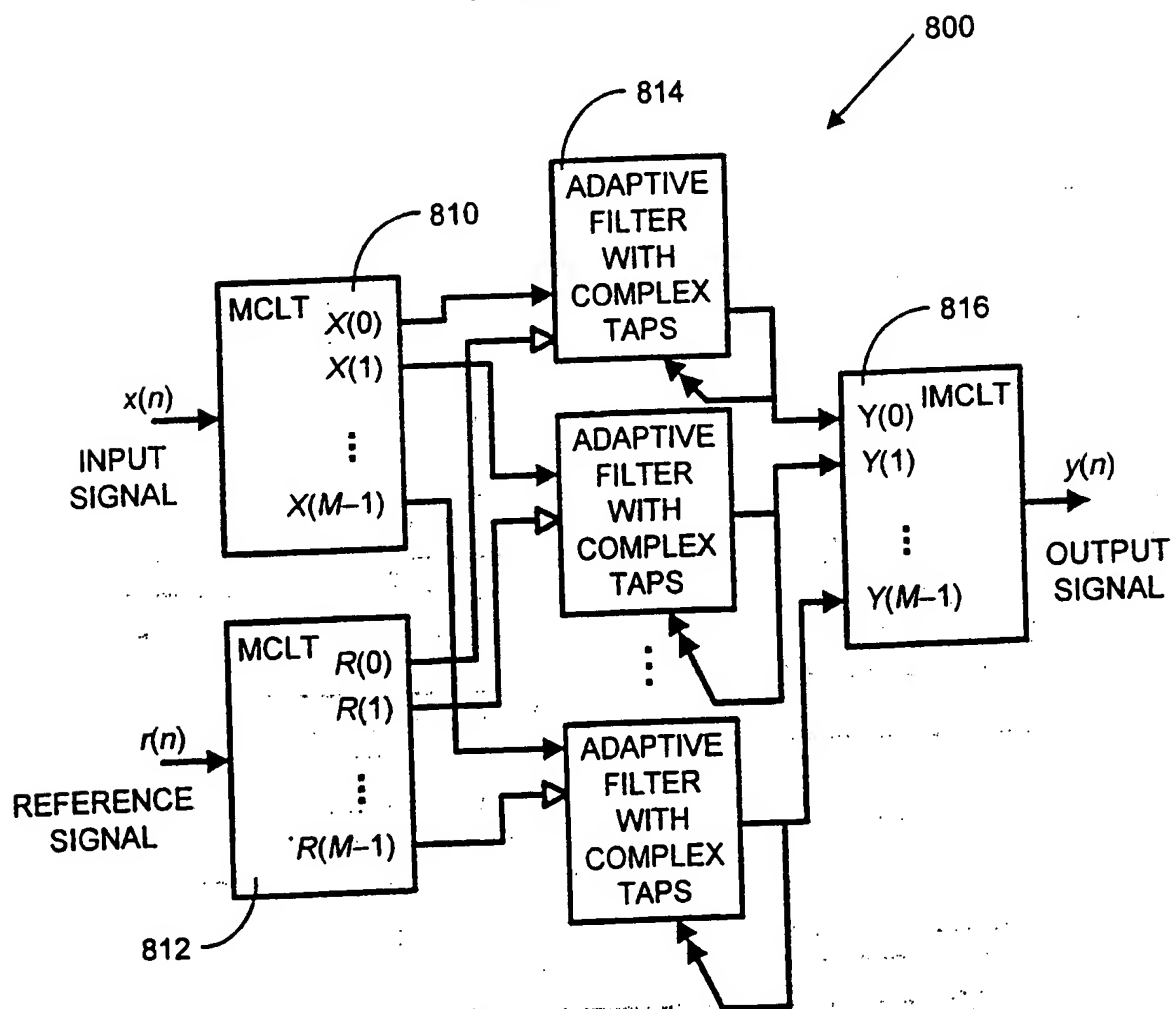


FIG. 7



FREQUENCY-DOMAIN  
MCLT-BASED  
ADAPTIVE FILTER

FIG. 8

9/13

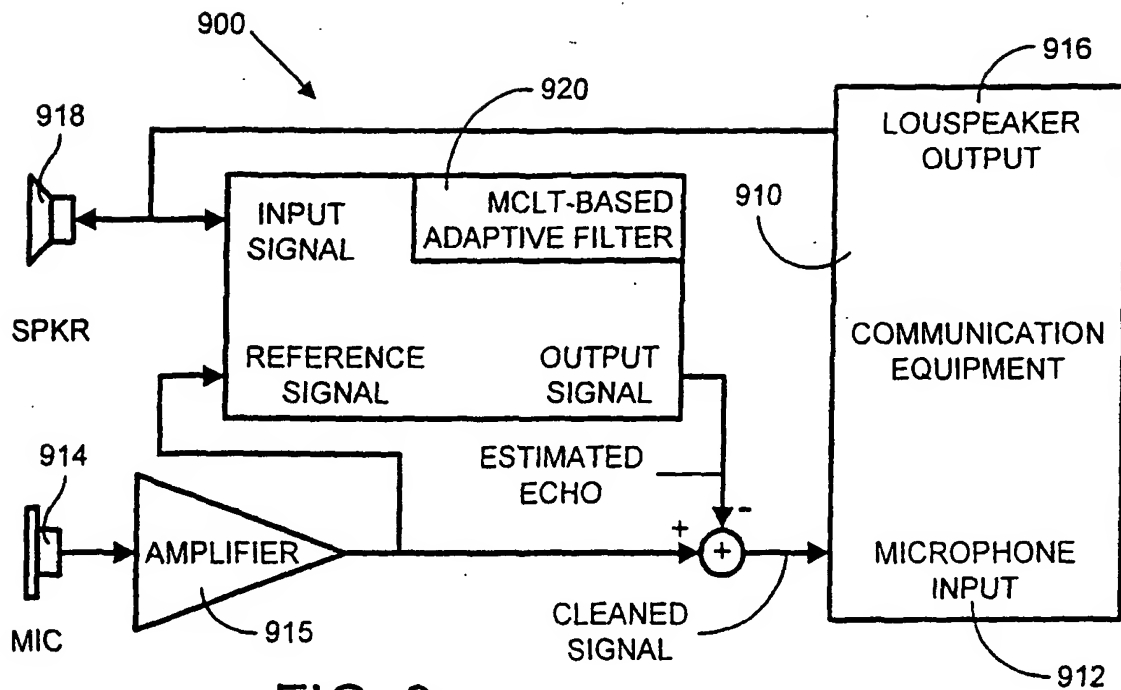


FIG. 9

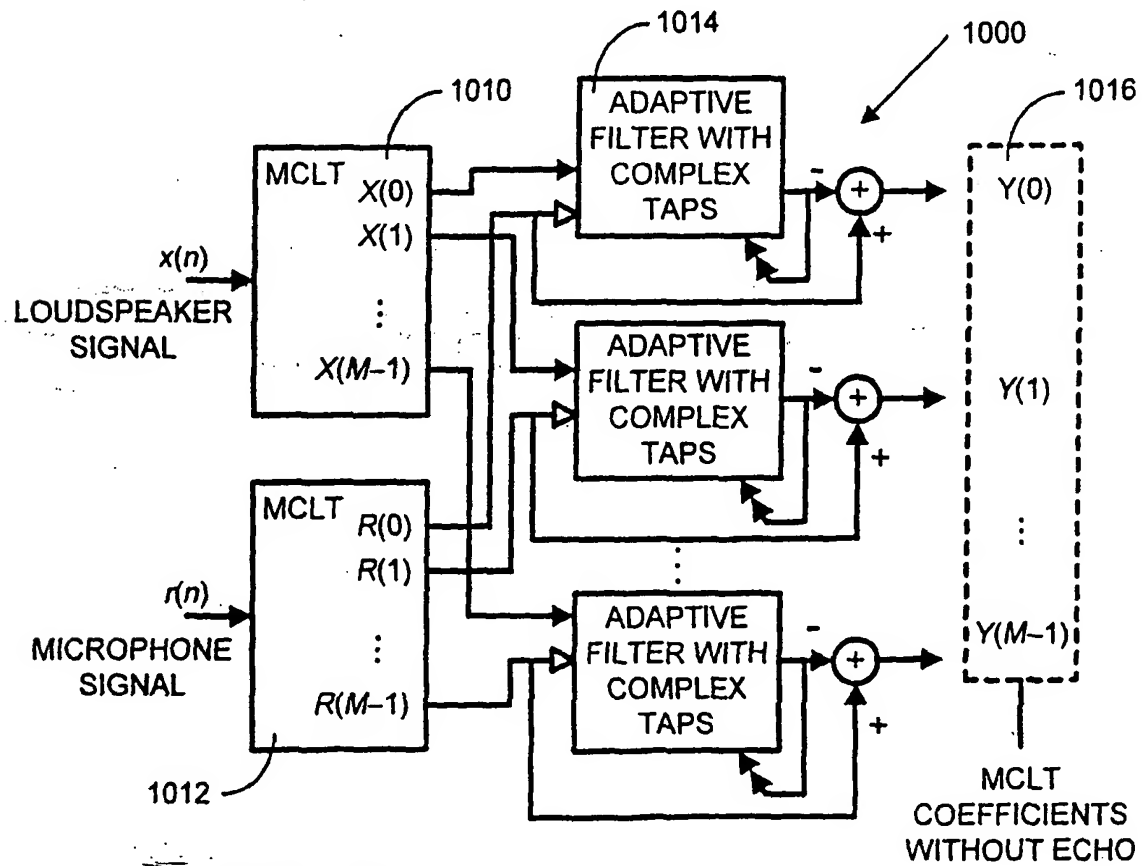


FIG. 10

10/13

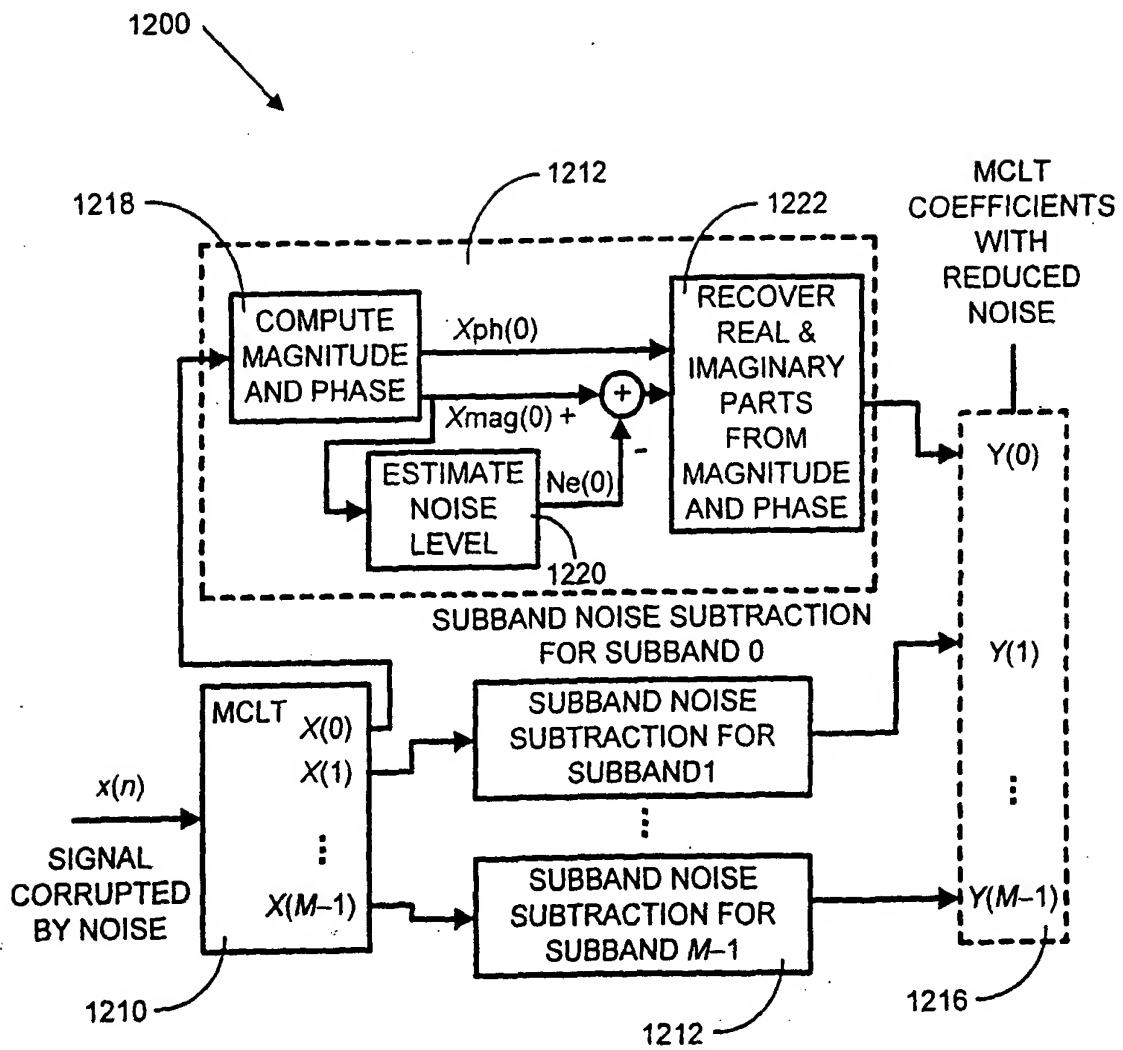


FIG. 12

11/13

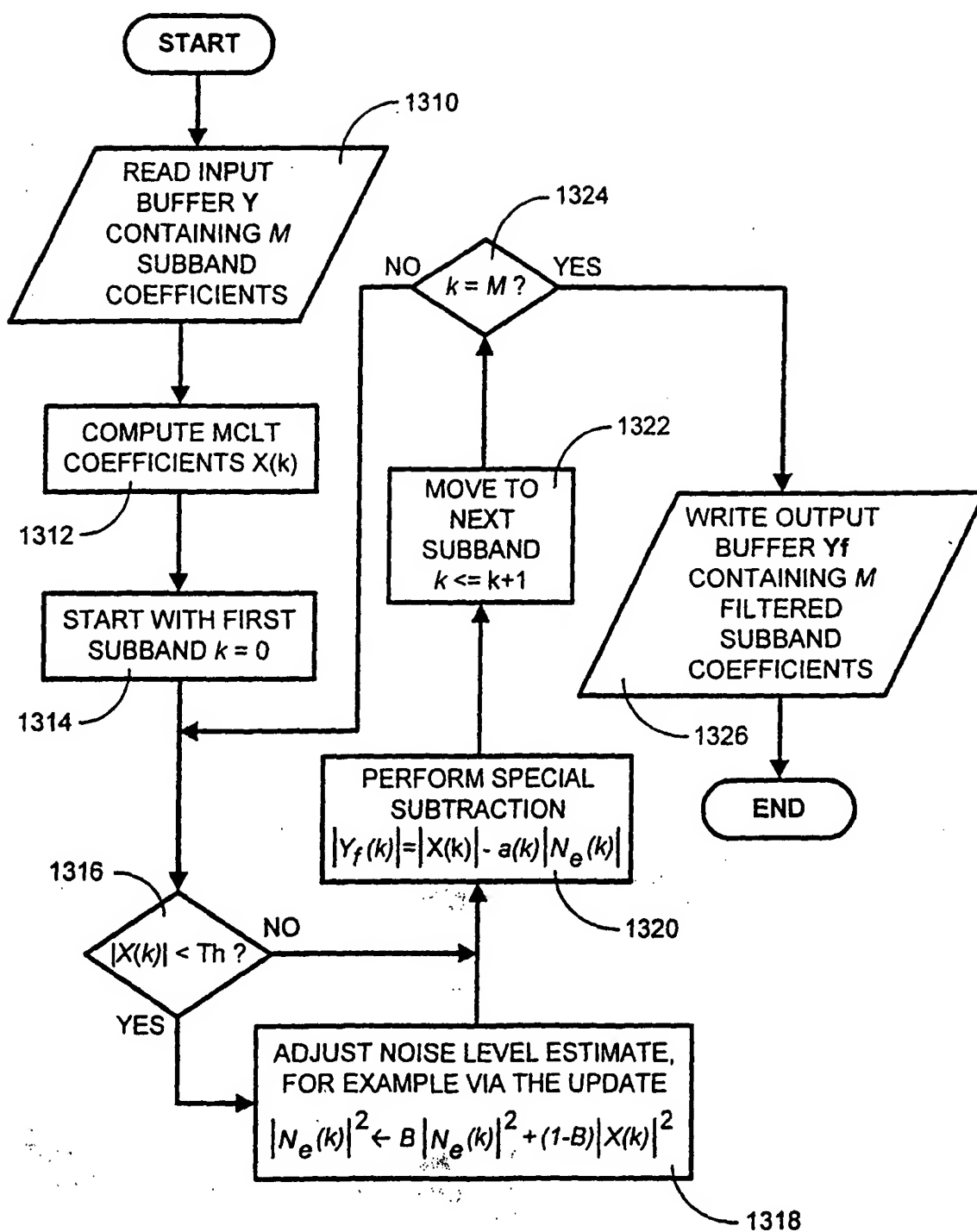


FIG. 13

12/13

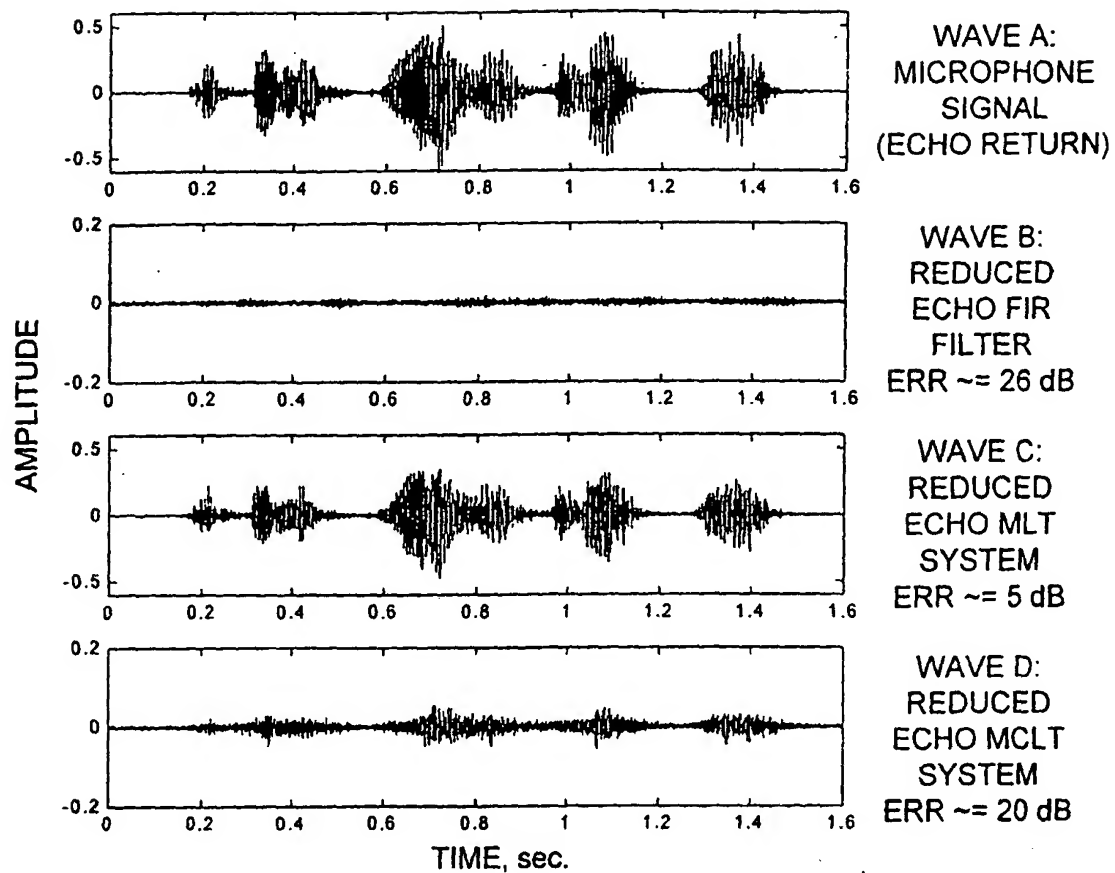


FIG. 11

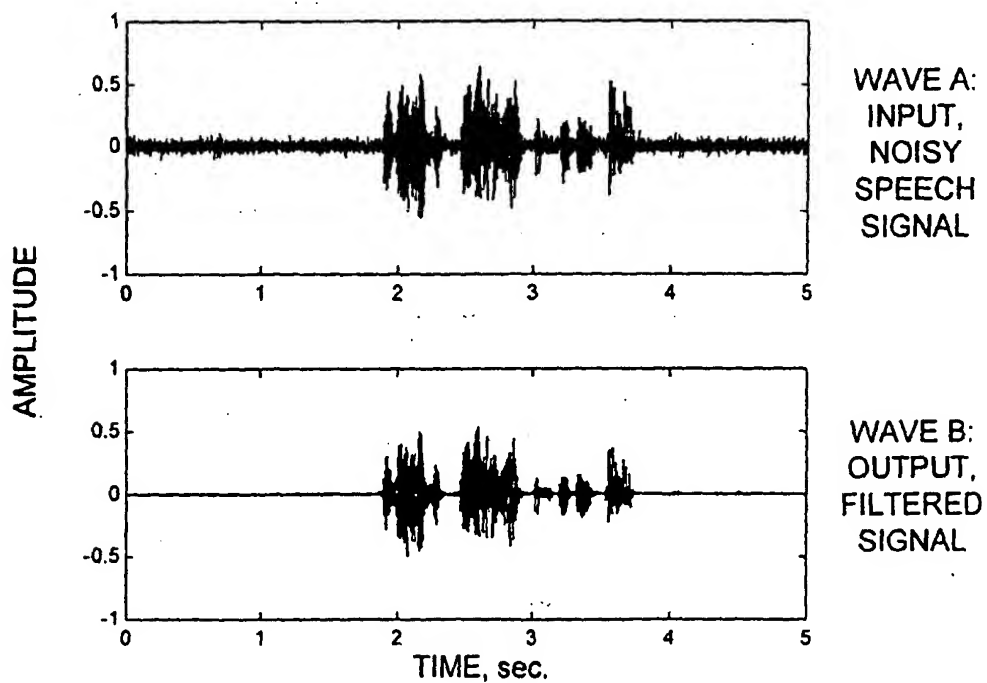


FIG. 14

13/13

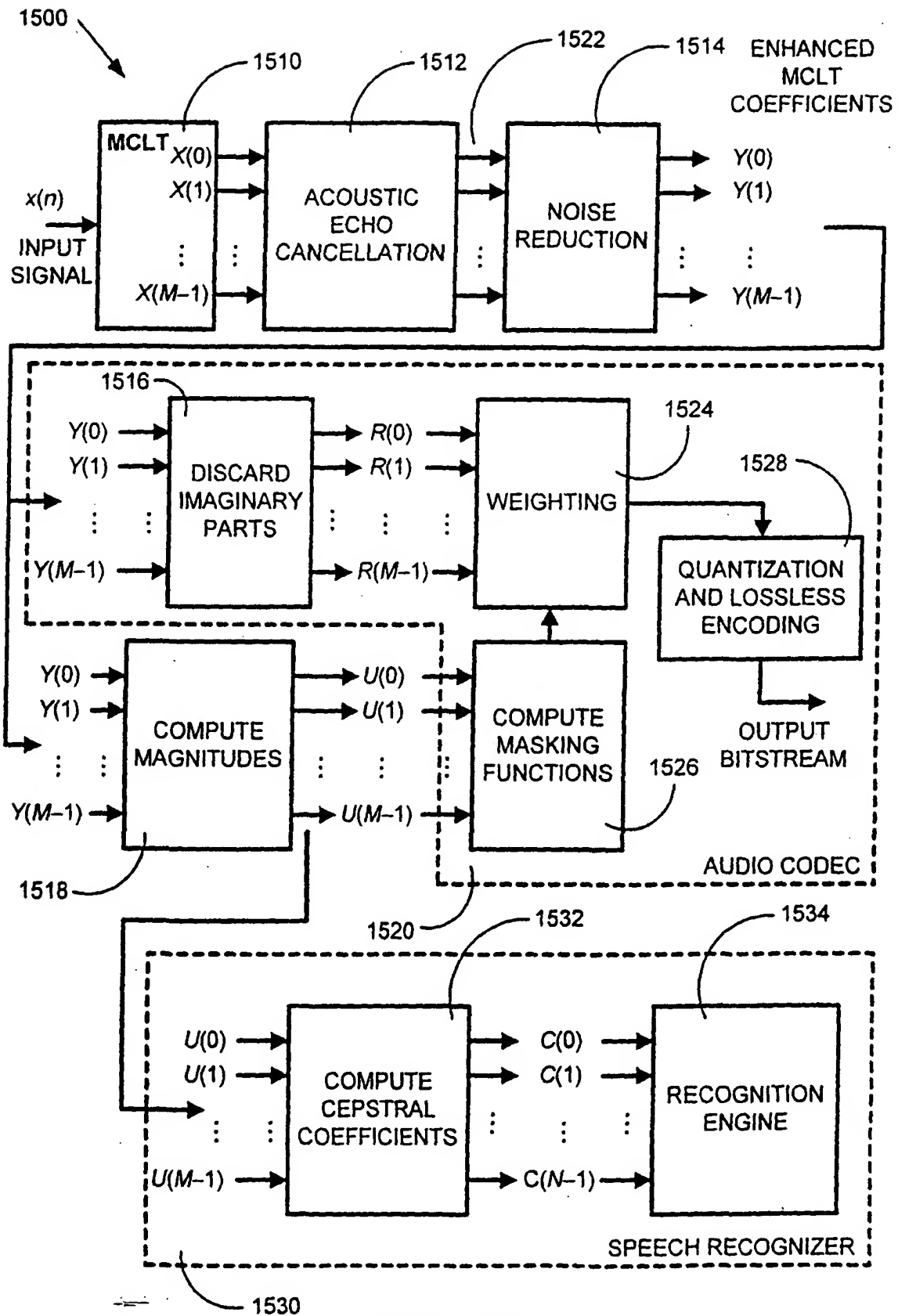


FIG. 15

# SP 24.6: A 500MHz 4Mb CMOS Pipeline-Burst Cache SRAM with Point-to-Point Noise Reduction Coding I/O

Kazuyuki Nakamura, Koichi Takeda, Hideo Toyoshima,  
Kenji Noda<sup>1</sup>, Hiroaki Ohkubo<sup>1</sup>, Tetsuya Uchida<sup>1</sup>,  
Toshiyuki Shimizu<sup>1</sup>, Toshiro Itani<sup>1</sup>, Ken Tokashiki<sup>1</sup>, Koji Kishimoto<sup>1</sup>

Microelectronics Research Labs/<sup>1</sup>ULSI Device Development Labs,  
NEC Corporation, Kanagawa, Japan

A secondary cache SRAM is an indispensable CPU partner in a high-performance system. The main objectives are 1) pipeline-burst operation, 2) 32b 500MHz (2GB/s) I/Os, and 3) point-to-point communication with a CPU, as well as shortened latency and reduced noise and power caused by high-speed, high-bandwidth I/O operation.

A pre-fetched pipeline scheme enables the cycle time for an internal memory core (I-cycle) to be extended by N times that of an external bus cycle (E-cycle) [1]. This is modified to an SRAM to achieve both 4b pipeline-burst cache operation and 500MHz I/O frequency. In this case, I-cycle time of 8ns is four times of E-cycle time (2ns). Figure 1 shows a timing diagram of the scheme, a pre-fetched pipeline-burst (PPB) with double late-write buffers (DLWBs). A FIFO-style DLWBs and shifted write data stream from a CPU to four E-cycles later than the corresponding address(ⓐ) are provided to completely eliminate idle cycles on the data bus that would occur in a read after write (RAW) situation. In read operations, four 32b data for one burst read cycle are simultaneously pre-fetched from an SRAM core. During read, the last two write operations are in the buffer queue. The actual write of the data into the SRAM core is postponed to two write-I-cycles later (ⓑ). PPB with DLWB accomplishes 4b pipeline-burst operation without idle cycles, even in continuous RAW and WAR (write after read) situations.

A block diagram of the chip is shown in Figure 2. An 128b SRAM core is operated in an I-cycle of 8ns, while a 32b off-chip I/O is operated in an E-cycle of 2ns. In read, a read address is immediately applied to the SRAM core to pre-fetch 128b data for a burst read. A parallel-in serial-out (PISO) buffer converts the 128b I-cycle read-out data to a four 32b E-cycle data stream in the requested order. The data are transferred to the CPU through noise reduction (NR) encoding circuits every E-cycle. The DLWBs store two address-data pairs that have not been written to the SRAM core. When a read request matches the address stored in one of the DLWBs, the stored data in that DLWB is bypassed to data-out.

For 4 E-cycle latency for the head of the requested data, we reduced the access time for the internal 4Mb SRAM core to less than 2 E-cycle (4ns) by inserting a resetting period in the second half of the I-cycle. The resetting signal is applied at the source of the address decoder. Figure 3 shows the source-resetting scheme. Gate widths for pMOSFETs in NAND gates in the decoder can be halved by using source-resetting. This is because both the pMOSFETs can always help to pull-up the output node in the resetting period. The reduction in input capacitance for NAND gates decreases both delay time and power for decoder circuits. Figure 4 demonstrates the improvement in delay time distribution for the SRAM core. Gate-width reduction and bit-line self-equalization by word-line resetting reduce access time from 4.4ns to 3.5ns, in turn enabling 4-1-1-1 latency.

Since the data I/O bus between the secondary cache and CPU is connected point-to-point, are push-pull output drivers with impedance control circuits [2]. These drivers can achieve series-

termination by matching the impedance of output driver transistors to the transmission line impedance (about 50Ω). A custom BGA package is designed to precisely match the impedance of the data lines in the package.

A previously-proposed scheme to halve the number of simultaneous switchings of output drivers by adding a single redundant bit is extended to implement bi-directional data I/Os with no idle cycle 500MHz operations [3]. Figure 5a illustrates a scheme comparing the present data on bus and input data directly. Figure 5b is another approach, in which the procedure is divided into two steps, 1) input data is converted to a low-weight code, and 2) the reduced "1"s are converted to transitions. The former scheme is simpler. Switching of the data direction, however, should be considered to apply the coding to a bi-directional bus without idle-cycles. To send data immediately after receiving data, the data on the bus should be bypassed to an encoder circuit while receiving data (case ⓐ in Figure 5). The critical path for this bypass in one I/O cycle (E-cycle) is illustrated by bold lines in Figure 5. A majority voter circuit is responsible for most of the delay time [3]. Since this circuit is in the critical path in addition to the bus delay in the former scheme, the operation margin in one I/O cycle is degraded. The latter scheme, on the other hand, includes only one stage of XOR gate, enabling 500MHz bi-directional operation and has 0.7 and 0.3ns delay times for encoding and decoding, respectively. The noise level of the 36b coding I/O is equivalent to that of a conventional 16b I/O chip.

A chip micrograph is shown in Figure 6. This chip is fabricated with a 0.25μm triple-metal CMOS process. The 6-transistor memory cell is 3.04x4.2μm<sup>2</sup>. The die is 12.0x11.0mm<sup>2</sup>. The SRAM core is divided into four sub-arrays. Each array has 32kx32b cells, an 8b/32b serial-parallel converter, and an 8b/9b noise reduction coding I/O. This symmetrical layout inhibits the local concentration of the power and noise. The 500MHz data I/O region and 125MHz SRAM core area are physically separated. There are three power supply lines and grounds, one for the SRAM core (2.5V), one for the data I/O(2.5V), and one for the output transistors (1.0V). The interface level is 1.0V, and the reference voltage is 0.5V. Two reference resistors of 50Ω are also required to set up the impedance of push-pull off-chip driver transistors.

Figure 7 is a measured Shmoo plot. The data window at 500MHz is 1.2ns. Supply current at 500MHz is about 270mA. Estimated current dissipated in the bus is 40mA for 4cm between CPU and the chip, is only one-eighth that in conventional open-drain I/O.

## Acknowledgments:

The authors thank M. Yotsuyanagi, M. Fukuma, S. Ohya, K. Nakamura, Y. Murao and T. Kikkawa, for encouragement and suggestions, and F. Hayashi, S. Horiba, T. Takahashi, T. Yokoyama, K. Yoshida, T. Shinmura, and K. Koyanagi for assistance in chip fabrication.

## References:

- [1] Saeki, T., et al., "A 2.5ns Clock Access 250MHz 256Mb SDRAM with Synchronous Mirror Delay," ISSCC Digest of Technical Papers, pp. 374-375, Feb., 1996.
- [2] Pilo, H., et al., "A 300MHz 3.3V 1Mb SRAM Fabricated in a 0.5μm CMOS Process," ISSCC Digest of Technical Papers, pp. 148-149, Feb., 1996.
- [3] Nakamura, K., et al., "A 50% Noise Reduction Interface Using Low-Weight Coding," Symp. On VLSI Circuits, pp. 144-145, June, 1996.



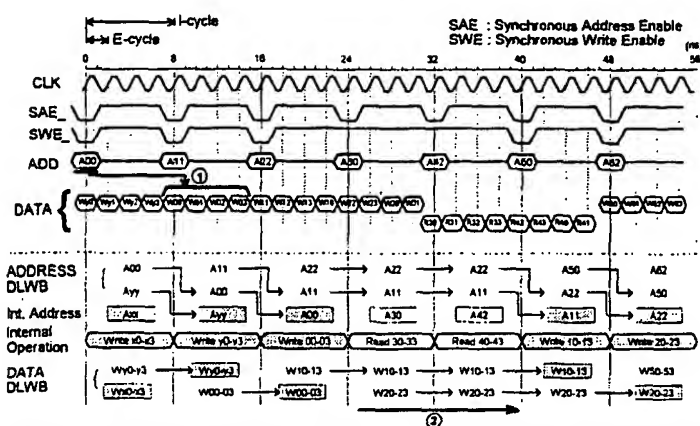


Figure 1: Timing diagram for 500MHz no-idle cycle PPB operation.

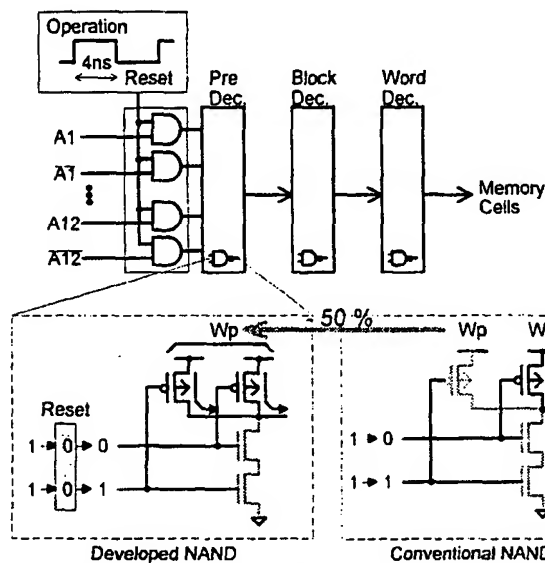


Figure 3: Gate width reduction by source-resetting scheme.

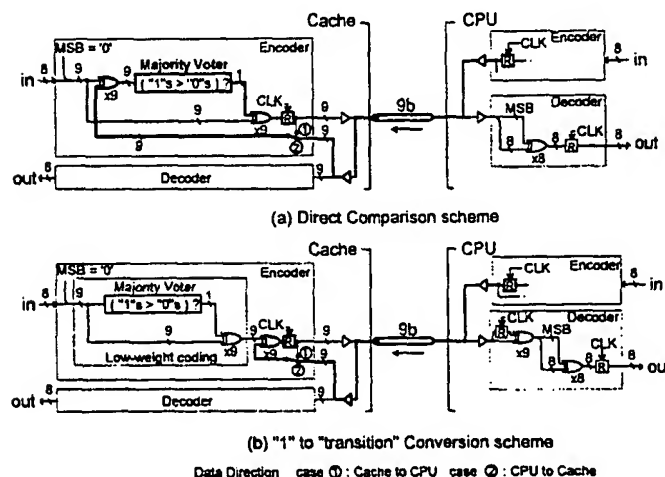


Figure 5: Bi-directional noise reduction coding circuit.  
Figure 6: See page 495.

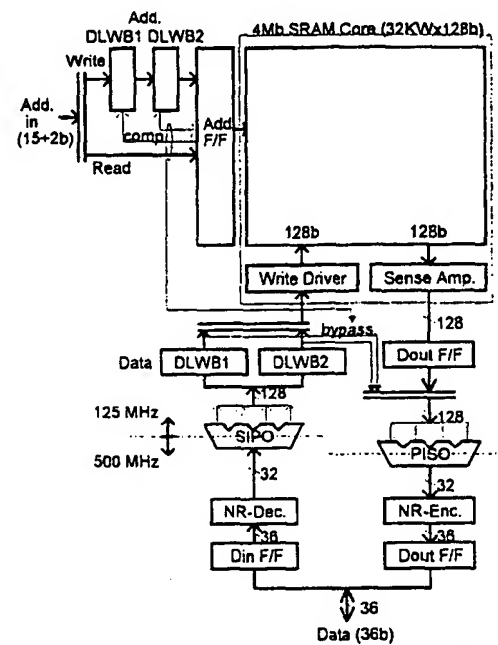


Figure 2: Block diagram with double-late-write-buffers (DLWBs).

a) Developed SRAM (Source-Resetting)

Decoder	Bitline → SAout	→ Dout	
1.4 ns	1.3 ns	0.8 ns	3.5 ns
Decoder	Bitline → SAout	→ Dout	
1.7 ns	1.9 ns	0.8 ns	4.4 ns

b) Conventional SRAM

Figure 4: SRAM core delay time comparison.

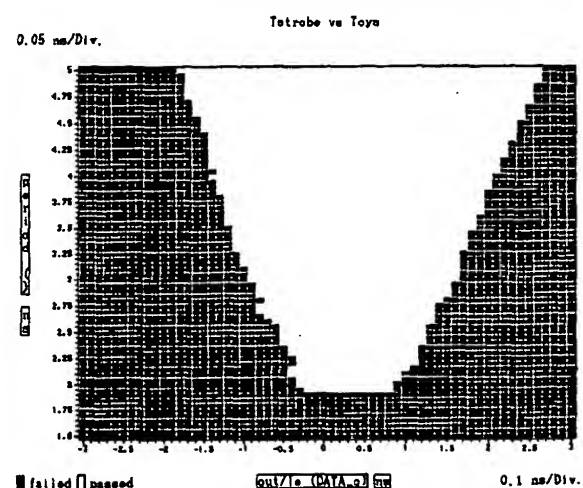


Figure 7: Shmoo plot (data output margin - cycle time).

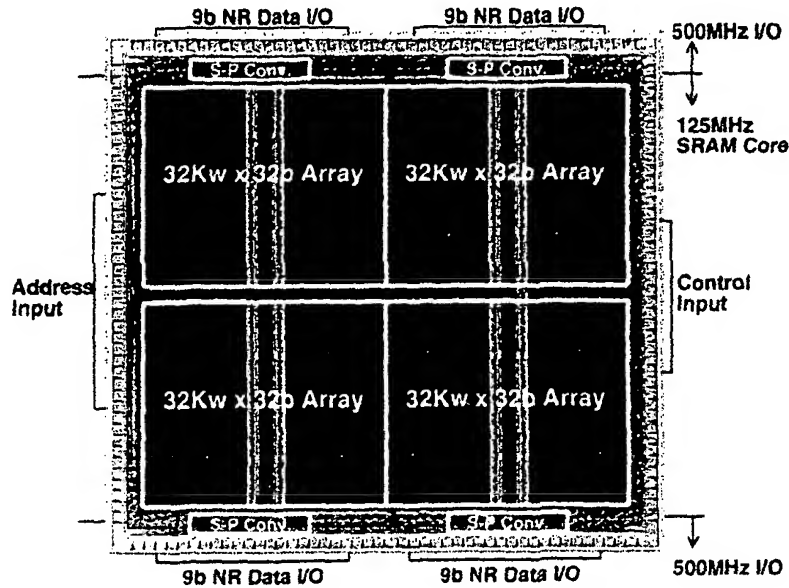


Figure 6: 4Mb SRAM chip micrograph.

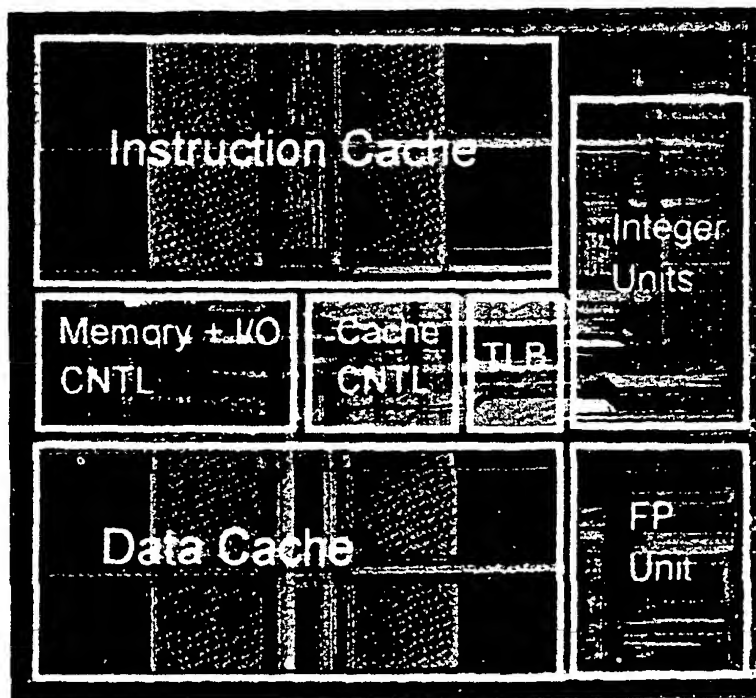


Figure 6: Chip micrograph.